УЧЕБНИК ДЛЯ ВЫСШЕЙ ШКОЛЫ

Л. Н. Ясницкий

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ



УЧЕБНИК ДЛЯ ВЫСШЕЙ ШКОЛЫ

Л. Н. Ясницкий

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Электронное издание

Рекомендовано УМО
по классическому университетскому образованию
в качестве учебника для студентов высших учебных заведений,
обучающихся по направлению ВПО 010300 «Фундаментальная
информатика и информационные технологии»



Москва
Лаборатория знаний
2016

УДК 004.8
ББК 32.813я73
Я82

Серия основана в 2009 г.

Рецензенты: зав. кафедрой информационных технологий
Пермского государственного национального исследовательского университета,
д-р физ.-мат. наук, проф. Е. К. Хеннер;
зав. кафедрой прикладной математики и информатики
Пермского государственного национального исследовательского университета,
д-р физ.-мат. наук, проф. С. В. Русаков;
профессор кафедры математического моделирования систем и процессов
Пермского национального исследовательского политехнического университета,
д-р техн. наук, проф. В. Ю. Столбов.

Ясницкий Л. Н.

Я82 Интеллектуальные системы [Электронный ресурс] : учебник / Л. Н. Ясницкий. — Эл. изд. — Электрон. текстовые дан. (1 файл pdf : 224 с.). — М. : Лаборатория знаний, 2016. — (Учебник для высшей школы). — Систем. требования: Adobe Reader XI ; экран 10".

ISBN 978-5-00101-417-1

В учебнике приведена история становления научной области «искусственный интеллект». Освещены основные направления ее развития и сферы применения; выполнено сопоставление трех основных стратегических подходов к созданию интеллектуальных систем: технологии экспертных систем, технологии нейронных сетей и технологии эволюционного моделирования. Изложены теоретические основы и даны примеры разработки интеллектуальных систем, а также примеры их применения для интеллектуального анализа данных в промышленности, экономике, бизнесе, психологии, социологии и других областях.

Книга является исчерпывающим руководством по освоению технологий создания интеллектуальных нейросетевых систем и их применению для решения широкого круга проблем, встречающихся во многих областях деятельности современного человека.

Для студентов высших учебных заведений технического профиля.

УДК 004.8
ББК 32.813я73

Деривативное электронное издание на основе печатного аналога: Интеллектуальные системы : учебник / Л. Н. Ясницкий. — М. : Лаборатория знаний, 2016. — 221 с. : ил. — (Учебник для высшей школы). — ISBN 978-5-906828-73-6.

ISBN 978-5-00101-417-1

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

В последнее время руководители системы высшего профессионального образования уделяют внимание изучению методов создания интеллектуальных систем, включая в учебные программы вузов дисциплины под названиями:

- «Интеллектуальные информационные системы»;
- «Системы искусственного интеллекта»;
- «Нейронные сети»;
- «Интеллектуальные системы»;
- «Интеллектуальные системы и технологии»;
- «Интеллектуальные технологии»;
- «Основы искусственного интеллекта»;
- «Прикладные методы искусственного интеллекта»;
- «Интеллектуальный анализ данных».

В чрезвычайно высокой востребованности указанных дисциплин автор этой книги многократно убеждался, сотрудничая со многими фирмами — разработчиками и пользователями IT-продукции, а также читая эти дисциплины во многих университетах. Как показывает опыт, изучение методов создания интеллектуальных систем обычно не ограничивается сдачей зачетов и экзаменов. Освоив теоретическую базу и технологии создания интеллектуальных систем, студенты продолжают их применять при выполнении курсовых, дипломных, магистерских, кандидатских и докторских работ. Уже после окончания вуза выпускники делают себе карьеру, создают свои собственные инновационные предприятия, разрабатывая и внедряя интеллектуальные информационные системы в экономике, в промышленности, в бизнесе, в политологии и социологии, в медицине и криминалистике, педагогике и психологии и т. д. Причем сфера их деятельности и внедрений постоянно расширяется, а успехи умножаются.

Как заметил известный польский специалист в области нейросетевых технологий Рышард Тадеусевич в своей последней книге[1)], «мода на применение [методов искусственного интеллекта] продолжается

[1)] Элементарное введение в технологию нейронных сетей с примерами программ / Пер. с польск. И. Д. Рудинского. — М.: Горячая линия — Телеком, 2011. — 408 с.

и даже расширяется... В некоторых кругах незнание нейронных сетей начинает трактоваться как светская бестактность! ... Восхищение нейронными сетями имеет вполне конкретные и важные причины. Главная предпосылка их огромной популярности — действительно прекрасные результаты, получаемые с помощью этого нового инструментария при решении многих задач, издавна считавшихся особо сложными».

Назовем несколько таких «прекрасных результатов», полученных студентами и аспирантами вузов и сотрудниками предприятий под руководством и при участии автора этой книги.

1. Разработана система диагностики авиационных двигателей, по точности и спектру обнаруживаемых дефектов превосходящая существующие инженерные методики[1)].
2. В 2006 г. была спрогнозирована победа Д. А. Медведева на президентских выборах 2008 г. Прогноз[2)] был опубликован в 2007 г. — за полтора года до выборов, когда личность Д. А. Медведева как политика еще была мало известна и его победа на выборах мало кем ожидалась, однако прогноз нейронной сети оказался верным.
3. В 2008 и 2010 гг., когда президент Д. А. Медведев находился на вершине своей популярности, были опубликованы работы[3)][4)], в которых искусственный интеллект прогнозировал ему снижение рейтинга, тогда как другому политику — В. В. Жириновскому, прогнозные кривые предсказали постепенный рост популярности, что и наблюдалось в последующие годы.
4. В 2013 г. выполнен и опубликован[5)] подтвердившийся впоследствии[6)] прогноз места в неофициальном командном зачете российских спортсменов на Зимней Олимпиаде-2014. Кроме того, были разработаны рекомендации по улучшению результативности

[1)] *Конев С. В., Сичинава З. И., Ясницкий Л. Н.* Применение нейросетевых технологий для диагностики неисправностей авиационных двигателей. Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2005. № 2. С. 43–47 с.

[2)] *Ясницкий Л. Н.* Интеллектуальные информационные технологии и системы. Пермь: Пермский университет, 2007. 271 с.

[3)] *Ясницкий Л. Н.* О возможностях применения методов искусственного интеллекта в политологии // Вестник Пермского университета. Серия: Политология. 2008. № 2. С. 147–155.

[4)] *Ясницкий Л. Н., Черепанов Ф. М.* О возможностях применения нейросетевых технологий в политологии // Нейрокомпьютеры: разработка, применение. 2010. № 8. С. 47–53.

[5)] *Ясницкий Л. Н., Павлов И. В., Черепанов Ф. М.* Прогнозирование результатов Олимпийских игр 2014 г. в неофициальном командном зачете методами искусственного интеллекта // Современные проблемы науки и образования. — 2013. — № 6; URL: **http://www.science-education.ru/113\nobreakdash-11206** (дата обращения: 25.12.2013).

[6)] **http://www.psu.ru/news/prognoz-uchenykh-pgniu-okazalsya-vernym**.

отдельных спортсменов[1]. Некоторые из этих рекомендаций были учтены тренерами и дали результат на Олимпиаде-2014[2].

5. В ходе разработки и исследований нейросетевой системы диагностики заболеваний сердечно-сосудистой системы методами искусственного интеллекта выявлены новые медицинские знания, в результате чего, после обсуждений с авторитетными врачами-экспертами, сделан вывод[3] о необходимости корректировки существующей практики профилактики и лечения кардиологических заболеваний.
6. На сайте Пермского отделения Научного совета РАН по методологии искусственного интеллекта **www.PermAI.ru** в разделе «Проекты» в свободном доступе выложены программы, позволяющие: прогнозировать вероятность банкротства российских банков; выявлять способности людей к научной деятельности и к руководящей деятельности; прогнозировать успешность карьеры студентов после окончания вуза; прогнозировать прочность семьи; выявлять предрасположенность людей к наркозависимости, к суициду, к насилию, к анорексии; ставить диагнозы заболеваний сердечно-сосудистой системы и выполнять прогнозирование их развития на ближайшие годы. Воспользовавшись указанными программами, читатели этой книги могут выбрать для себя сферу деятельности, занимаясь которой они добьются наибольших успехов; определять оптимальную для себя траекторию обучения в вузе; определять оптимальные для себя меры по снижению их предрасположенности к таким заболеваниям, как наркомания, суицид, анорексия; поставить себе диагнозы заболеваний сердечно-сосудистой системы и определить оптимальные для себя меры по излечению этих заболеваний; проверять насколько надежен банк, выбранный для хранения сбережений.

Являясь председателем Пермского отделения Научного совета по методологии искусственного интеллекта Российской академии наук, сотрудником нескольких IT-фирм, профессором пяти университетов,

---

[1] *Ясницкий Л. Н., Внукова О. В., Черепанов Ф. М.* Прогноз результатов Олимпиады-2014 в мужском одиночном фигурном катании методами искусственного интеллекта // Современные проблемы науки и образования. — 2014. — № 1; URL: **http://www.science-education.ru/115-11339** (дата обращения: 25.12.2013).

[2] В указанной работе фигуристу Е. В. Плющенко прогнозировалось 2-е место на Олимпиаде-2014. Однако в этой же работе ему было рекомендовано сбавить вес, с чем и согласился его тренер А. Н. Мишин (см. **http://www.poisknews.ru/theme/infosphere/8916/**). В результате в командных соревнованиях на Олимпиаде-2014 Е. В. Плющенко показал блестящий результат, благодаря чему команда получила золотую медаль.

[3] *Yasnitsky L. N., Bogdanov K. V., Cherepanov F. M., Makurina T. V., Dumler A. A., Chugaynov S. V., Poleschuk A. N.* Diagnosis and Prognosis of Cardiovascular Diseases on the Basis of Neural Networks // Biomedical Engineering. 2013. T. 47. № 3. C. 160–163.

автор книги может долго перечислять достижения своих бывших студентов и аспирантов, ныне успешных бизнесменов, инженеров, научных работников, педагогов. Своей же главной заслугой он считает популяризацию этой достаточно сложной научной дисциплины — создание простой и эффективной методики ее преподавания, частью которой является данная книга.

Книга имеет хорошо зарекомендовавший себя прототип — учебное пособие Л. Н. Ясницкого «Введение в искусственный интеллект», первоначально выпущенное Пермским государственным университетом в 2001 г., а затем переизданное в Москве в 2005, 2008 и 2010 гг. В настоящее время эти книги используются в учебном процессе многих вузов и об их высокой востребованности свидетельствует большое количество цитирований, зафиксированных eLIBRARY, Академия GOOGLE и др.

БЛАГОДАРНОСТИ

Автор использовал рукописные материалы из библиотеки первого заведующего кафедрой прикладной математики Пермского государственного университета (ныне Пермский государственный национальный исследовательский университет) Юрия Владимировича Девингталя, подаренной университету Валентиной Васильевной Девингталь. Кроме того, в книге приводится информация, любезно предоставленная автору специалистами в области искусственного интеллекта: Р. П. Абусевым, И. А. Грибановым, В. А. Игошиным, В. А. Краснобаевым, М. А. Марценюком, С. И. Чуприной. Компьютерная реализация идеи социально-генетического алгоритма и его апробация выполнены совместно с Д. И. Мурашовым. Принципы построения нейросетевой системы диагностики авиационных двигателей разработаны совместно с начальником отдела диагностики АО «Авиадвигатель» В. Ф. Халиуллиным, а ее программную реализацию выполнили З. И. Сичинава и С. В. Конев. Принципиальные основы создания системы кардиодиагностики разрабатывались совместно с врачом — консультантом Пермского кардиологического центра Ю. К. Филоненко и доцентом Пермского государственного медицинского университета им. академика Е. А. Вагнера А. А. Думлером, а ее программная реализация и компьютерные эксперименты выполнены М. А. Таначевой и А. Н. Дворяниновой, Ф. М. Черепановым К. В. Богдановым, Т. В. Макуриной, А. С. Плюсниным. Опытно-промышленные испытания, окончательная доработка и внедрение кардиодиагностической системы осуществлялись под руководством директора ЗАО «Информационно-вычислительные сети» А. Н. Полещука при финансовой поддержке правительства Пермского края.

В разработке интеллектуальных информационных систем в областях экономики, бизнеса, политологии, социологии, педагогики, медицины и других принимали участие студенты, магистранты, аспиранты и сотрудники практически всех пермских вузов.

Многие нейросетевые модели выполнены с использованием специально созданного для этих целей программного инструментария — «Нейросимулятора 5.0», явившегося результатом многолетнего сотрудничества автора с Ф. М. Черепановым. Совместно с ним же создан лабораторный практикум по искусственному интеллекту и интеллектуальному анализу данных, размещенный на сайте **www.LbAi.ru**.

Автор выражает свою признательность руководителям Научного Совета РАН по методологии искусственного интеллекта академику В. Л. Макарову и профессору Д. И. Дубровскому за внимание и поддержку своего пермского филиала.

Автор чрезвычайно благодарен своим наставникам и коллегам по редакционной коллегии журнала «Нейрокомпьютеры: разработка, применение» А. И. Галушкину, А. В. Чечкину, Л. С. Куравскому, В. В. Борисову, А. В. Савельеву.

Особую благодарность автор выражает Российскому фонду фундаментальных исследований (РФФИ) за выделение гранта РФФИ № 16-01-00164 для подготовки рукописи книги.

Всем, кто предоставил информацию, принял участие в разработке интеллектуальных систем и в подготовке материалов книги, всем прямым и косвенным участникам Пермской научной школы искусственного интеллекта автор выражает искреннюю благодарность.

Глава 1

ПРОШЛОЕ И НАСТОЯЩЕЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

1.1. ИСТОРИЧЕСКИЙ ОЧЕРК

Человек — это самый сложный из доступных для нашего восприятия объект, а способность мышления — его главное свойство — атрибут. *Искусственный интеллект* — это наука, целью которой является изучение и моделирование атрибута человека — *мышления*.

Какова природа мышления? Какие процессы происходят в нашем организме, когда мы думаем, чувствуем, видим, понимаем? Возможно ли в принципе понять, как работает наш мозг, и заставить мыслить неживую материю? На протяжении тысячелетий человек задавался этими вопросами, но до сих пор мы не можем ответить на них с полной определенностью.

Попытки понять и формализовать механизм мышления человека имеют давнюю историю. Еще Платон (IV в. до н. э.), Аристотель (IV в. до н. э.), Рене Декарт (1596–1650), Бенедикт Спиноза (1632–1677), Готфрид Вильгельм Лейбниц (1646–1716), Джордж Буль (1815–1864) и многие другие исследователи на уровне современных им знаний стремились описать мышление как совокупность некоторых элементарных операций, правил и процедур. В полном объеме формальная аксиоматизация логических рассуждений представлена в работах таких великих философов, как Фридрих Людвиг Готлоб Фреге (1848–1925), Бертран Артур Уильям Рассел (1872–1970), Курт Гёдель (1906–1978).

Первую, зафиксированную в истории попытку создания машины, моделирующей человеческий разум, связывают с именем средневекового испанского рыцаря, поэта, философа, богослова, алхимика, изобретателя Раймунда Луллия (1235–1315). Представляет огромный интерес сама личность этого человека (рис. 1.1). Любимец короля, дуэлянт и повеса, как о нем пишут историки, рыцарь Раймунд Луллий в возрасте 24 лет вдруг неожиданно отказывается от светских развлечений и идет в монастырь, чтобы стать мудрецом. Согласно легенде, его решение продиктовано весьма благородной целью — постичь науки и с их помощью излечить от тяжелой болезни свою даму сердца.

К сожалению, истории неизвестно, удалось ли Луллию достичь своей благородной цели. Известно только, что в возрасте 80 лет во время жаркого спора его насмерть забили камнями оппоненты — те, кто были не согласны с учением Луллия…

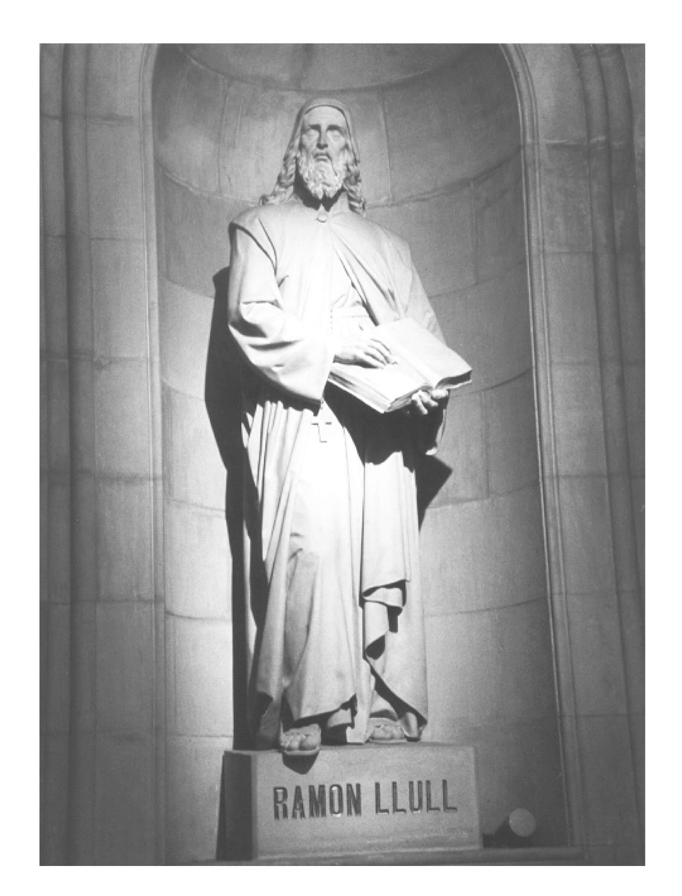


Рис. 1.1. Статуя Раймунда Луллия в Барселонском университете

Во времена Луллия, а это был XIII в., ученые занимались поиском неких универсальных понятий и истин, которые, будучи связанными между собой, давали бы общую картину мироздания, а значит, и ответы на все интересующие человечество вопросы. Это был век философов-мудрецов, астрологов и алхимиков, занятых поисками философского камня.

Развивая традиции ученых своего времени, Луллий сконструировал механическое устройство, состоящее из системы кругов (рис. 1.2), имеющих возможность вращаться. Каждый круг был поделен на секторы, окрашенные в разные цвета и помеченные латинскими буквами. Круги соединялись друг с другом и приводились во вращение. В результате получались различные сочетания символов и цветов — так называемая «формула истины».

Машины Луллия могли работать в различных предметных областях и давать ответы на всевозможные вопросы, составлять гороскопы,

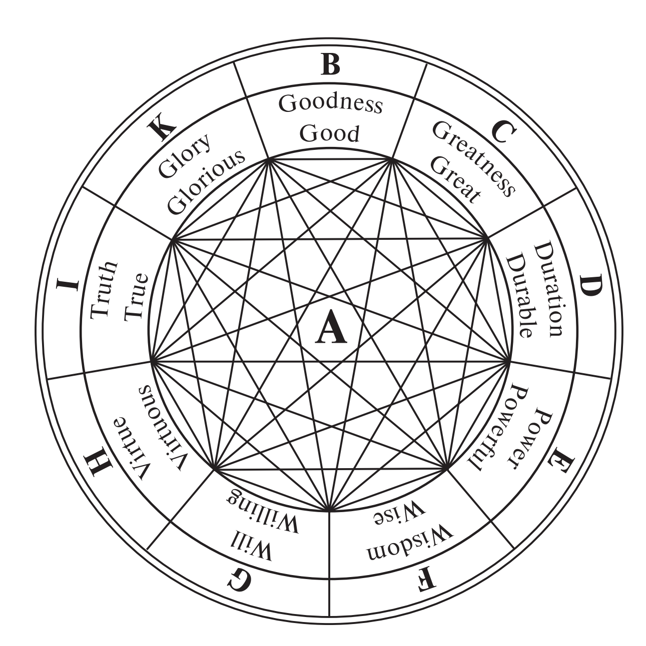


Рис. 1.2. Один из кругов логической машины Луллия

ставить диагнозы болезней, делать прогнозы на урожай. В наиболее позднем варианте машина Луллия состояла из 14 кругов, размеченных буквами и раскрашенных в различные цвета, которые символизировали различные понятия, элементы, стихии, субъекты и объекты знания. Круги приводились в движение системой рычагов. Поворачиваясь, они могли образовать около 18 квадриллионов ($18 \cdot 10^{15}$) разнообразных сочетаний буквенных и цветовых «истин». Запросы в машину вводились с помощью поворота внутреннего круга, на котором было начертано девять вариантов вопросов: «Что?», «Почему?», «Из чего?», «Сколько?», «Каким образом?», «Где?», «Когда?», «Какое?», «Которое из двух?».

По существу, машина Луллия представляла собой механическую экспертную систему, наделенную базой знаний, устройствами ввода и вывода, естественным языком общения. Свести к логическим операциям если не все знания о мире, то хотя бы часть из них, а затем поручить не человеческому мозгу, а механическому устройству процедуру вывода «формул знания», следующих из накопленной базы знаний, — эта идея искусственного интеллекта. Впервые она была высказана и реализована средневековым рыцарем Раймундом Луллием, прожила семь веков и достигла в наши дни своего расцвета и триумфа.

В 40-х гг. XX в. с появлением электронно-вычислительных машин искусственный интеллект обрел второе дыхание. Произошло выделение искусственного интеллекта в самостоятельное научное направление.

Сам термин «искусственный интеллект» (artificial intelligence) был предложен в 1956 г. на семинаре с аналогичным названием, проводимом в Стэнфордском университете (США).

С тех пор история искусственного интеллекта представляла собой постоянные споры и метания между двумя крайностями — оптимизмом и пессимизмом. Интересны знаменитые предсказания американского экономиста и социолога, исследователя в области теории управления, моделирования социальных процессов, нобелевского лауреата Герберта Саймона, сделанные в 1957 г. Приведем некоторые из этих предсказаний:

- в ближайшее десятилетие ЭВМ завоюет титул чемпиона мира по шахматам;
- в пределах десяти лет ЭВМ откроет и сумеет доказать важную новую математическую теорему;
- в десятилетний срок большинство теорий в области психологии примет вид программ для вычислительной машины.

Сейчас, спустя более полувека, мы видим, что предсказания Г. Саймона постепенно сбываются, что он ошибался только в сроках. Мы также можем отметить, что эйфория вокруг искусственного интеллекта имела как положительные, так и отрицательные последствия. С одной стороны, она стимулировала интерес общественности к новому научному направлению, выразившийся в выделении крупных грантов правительством США. С другой стороны, искусственный интеллект стал объектом весьма резкой критики более «трезвомыслящих» ученых. Мы также знаем, к каким тяжелым последствиям привела эта критика в СССР, когда за решение проблем научных дискуссий взялся государственный репрессивный аппарат: кибернетика была объявлена «буржуазной лженаукой», а те, кто ей занимался, — «врагами народа».

Вскоре после признания искусственного интеллекта самостоятельной отраслью науки произошло его разделение на два основных направления: *нейрокибернетику* и *кибернетику «черного ящика»*. Подход к созданию интеллектуальных систем, используемый в нейрокибернетике, иногда называют *низкоуровневым*, или *восходящим*, а в кибернетике черного ящика — *высокоуровневым,* или *нисходящим*.

Основную идею нейрокибернетики можно сформулировать следующим образом: «Единственный объект в природе, способный мыслить, — это человеческий мозг, поэтому любое мыслящее устройство должно быть обязательно выполнено по образу и подобию человеческого мозга, воспроизводить его структуру и его принцип действия». Таким образом, нейрокибернетика занимается моделированием структуры мозга и его деятельности.

Как известно, мозг человека состоит из большого количества взаимосвязанных нервных клеток — *нейронов*. Поэтому усилия нейрокибернетиков сосредоточены на разработке элементов, подобных нейронам,

и объединении этих элементов в системы — *нейросети* и *нейрокомпьютеры*. Первые нейросети и нейрокомпьютеры были созданы американскими учеными — Уорреном Мак-Каллоком, Уолтером Питтсом и Фредериком Розенблаттом в конце 1950-х гг. Это были устройства, моделирующие человеческий глаз и его взаимодействие с мозгом. Устройства умели распознавать буквы алфавита.

В отличие от нейрокибернетики кибернетика «черного ящика» не придает значения структуре и принципу действия мыслящего устройства. Главное, чтобы оно адекватно моделировало самый высокий уровень деятельности мозга — уровень интеллектуальных функций. Это направление искусственного интеллекта ориентировано на поиски алгоритмов решения интеллектуальных задач с использованием существующих компьютеров независимо от их аппаратной базы.

Поставив перед собой задачу моделирования функций мозга, представители этого научного направления столкнулись с серьезной проблемой. Оказалось, что, несмотря на многовековую историю исследований, ни одна из существующих наук (философия, психология, лингвистика и др.) не смогла предложить сколько-нибудь конкретный алгоритм человеческого мышления. Поэтому программистам пришлось создавать собственные модели мышления.

В конце 50-х гг. XX в. появилась *модель лабиринтного поиска*. Согласно этому подходу, решение интеллектуальной задачи выполнялось путем перебора огромного количества вариантов, которое представлялось в виде движения по лабиринту. В настоящее время модель лабиринтного поиска признается тупиковой и имеет ограниченное использование в игровых компьютерных программах.

В начале 1960-х гг. началась эпоха *эвристического программирования*. Как писал автор этого термина, американский математик П. Пойа, «цель эвристики — исследовать методы и правила, как делать открытия и изобретения».

Это очень сложная проблема. Дело в том, что Архимед, выпрыгнувший из ванны с криком «Эврика!», никому не объяснил, как он догадался, что тело, погруженное в жидкость, теряет в своем весе ровно столько, сколько весит вытесненный им объем воды. Ньютон открыл свой закон всемирного тяготения, наблюдая за падением яблока. Менделеев увидел принцип построения Периодической таблицы во сне. Поэтов и музыкантов вдохновляют на творческие поиски возвышенные чувства, разобраться в которых в принципе невозможно.

Чтобы понять механизмы творческого мышления, авторы эвристического подхода провели эксперимент. Была отобрана группа студентов, не знакомых с математической логикой. Каждый студент должен был самостоятельно доказать одну или несколько теорем из учебника, не заглядывая в него. При этом он должен был рассуждать вслух, делать любые записи, прекращать работу, если становилось ясно, что

выбран неверный путь, и начинать все сначала. Обработав магнитофонные записи, выкладки и черновики студентов, программисты нашли в них *эвристики* — некие правила, которыми пользовались студенты, доказывая теоремы. Эти правила не имели строгих математических обоснований, и их использование не всегда приводило к желаемому результату. Однако программа, в которую заложили эвристики студентов, смогла доказать все теоремы, какие имелись в учебнике, и сформулировать дополнительно еще две теоремы, которых не хватало до полной логической завершенности курса! Теперь эту программу, известную под названием «Логик-теоретик», принято считать родоначальницей *эвристического программирования*.

Сегодня инструментами нисходящего подхода к созданию интеллектуальных систем являются законы формальной логики, теории множеств, графов, семантических сетей и другие достижения науки в области дискретных вычислений. Ясно, что успехи нисходящего подхода связаны с развитием возможностей ЭВМ и искусства программирования, т. е. с успехами комплекса научных исследований, который называют «компьютерными науками». В противоположность этому успехи восходящего подхода (от низкого уровня к высокому) связаны с успехами наук о человеке.

Оба подхода к созданию интеллектуальных систем связаны с моделированием, однако в первом случае мы имеем дело с моделированием феноменологическим, имитационным, а во втором случае — с моделированием структурным.

Серьезный прорыв в практических приложениях нисходящего подхода к созданию интеллектуальных систем произошел в середине 1970-х гг., когда, отказавшись от поисков универсального алгоритма мышления, программисты начали моделировать конкретные знания специалистов-экспертов. Открылось новое направление искусственного интеллекта — *экспертные системы*.

С появлением экспертных систем бизнес в сфере интеллектуальных информационных технологий впервые становится рентабельным. С середины 1980-х гг. искусственный интеллект превратился в одно из наиболее привлекательных в коммерческом отношении направлений компьютерной индустрии. Растут ежегодные капиталовложения, создаются промышленные и военные экспертные системы.

Однако победа, одержанная приверженцами нисходящего подхода в 1980-х гг., оказалась неокончательной. Разобравшись в причинах своих неудач (подробно об этом см. гл. 4) и выполнив ряд серьезных фундаментальных исследований, ученые-нейрокибернетики вышли на рынок программной продукции с *нейронными сетями*. В результате 1990-е гг. ознаменовались приходом искусственных нейронных сетей в бизнес, где они показали свою реальную эффективность при решении многих практических задач. В последнее время наряду с указанными

выше двумя стратегическими подходами к проблеме моделирования мышления и интеллектуальных систем активно развивается третий, называемый *эволюционным*. Смысл этого подхода состоит в том, что процесс моделирования человеческого мозга заменяется моделированием процесса его эволюции. Здесь широко используются господствующие в живой природе механизмы естественного отбора и генетического наследования. Природные эволюционные процессы, описываемые теорией Чарльза Дарвина, кладутся в основу поведения простейших сущностей — *интеллектуальных агентов*, не имеющих тел, но способных чувствовать, действовать, бороться за существование, размножаться, развиваться и жить, как это происходит в физическом мире.

В заключение отметим, что исследования по искусственному интеллекту всегда находились на переднем крае информатики. Многие воспринимающиеся ныне вполне обычными программные средства, усовершенствованные системы программирования, операционные системы, текстовые редакторы, системы распознавания образов, ранее рассматривались как разработки искусственного интеллекта. Теории, идеи и методы искусственного интеллекта всегда привлекали и, вероятно, еще долго будут привлекать внимание тех, кто стремится расширить область применения и возможности компьютеров, сделать их более «дружественными», более похожими на разумных существ, с которыми приятно общаться и работать.

1.2. ОСНОВНЫЕ СТРАТЕГИИ

Искусственный интеллект — это научная отрасль, занимающаяся исследованием и моделированием естественного интеллекта человека. Естественный интеллект человека является очень сложным объектом исследований, и его моделирование осуществляется на разных уровнях абстрагирования [33]. Можно выделить три таких уровня, которым соответствуют три основные *стратегии* создания интеллектуальных систем (рис. 1.3):

1) технологии экспертных систем (высокоуровневая стратегия);
2) нейросетевые и нейрокомпьютерные технологии (низкоуровневая стратегия);
3) технологии эволюционного моделирования.

Самому высокому уровню абстрагирования соответствуют *технологии экспертных систем* — систем, основанных на явных знаниях о предметной области. Согласно этим стратегиям, знания о предметной области получают от специалистов-экспертов или из других информационных источников. Именно человек-эксперт, обладающий знаниями о предметной области, способен действовать на самом высоком уровне. Его знания формализуются и закачиваются в программный комплекс — экспертную систему, которая на основе этих знаний дела-

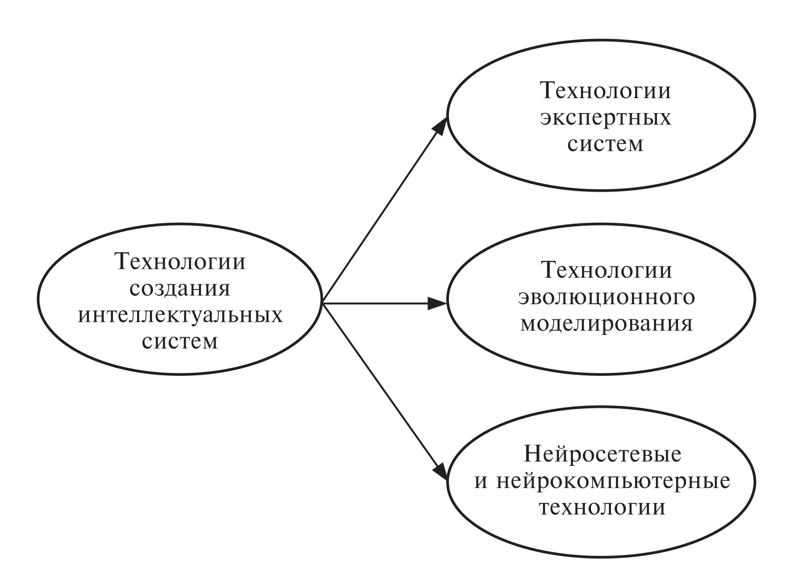


Рис. 1.3. Основные стратегии создания интеллектуальных систем

ет логически обоснованные полезные выводы, тем самым моделируя интеллект человека-эксперта. Именно эту стратегию впервые применил в XIII в. испанский ученый Раймунд Луллий. Именно с нее берет начало область науки, называемая искусственным интеллектом.

На самом нижнем уровне абстрагирования находятся *нейрокомпьютерные* и *нейросетевые технологии*. Согласно этой стратегии, строится модель, учитывающая структуру мозга, состоящего из множества нейронов, соединенных нервными волокнами. Знания поступают в такую модель в ходе ее обучения на специально подобранных примерах, характеризующих предметную область. Так же как и в мозге, знания хранятся в неявной форме — кодируются в виде множества *сил синаптических связей* (об этом пойдет речь в гл. 4), имитирующих электропроводность межнейронных соединений. Основоположниками этой стратегии, появившейся в середине XX в., являются американские ученые У. Мак-Каллок, В. Питтс и Ф. Розенблатт.

Исторически первой была модель интеллекта, основанного на явных знаниях (технология экспертных систем), которая безраздельно господствовала с XIII в. и до середины XX в. Возникновение в 1940-х гг. альтернативной стратегии и появление первого нейрокомпьютера, названного *персептроном*, равно как и первые его успехи, были встречены с большими надеждами и энтузиазмом, который, впрочем, продержался недолго. К середине 1980-х гг. подавляющее большинство интеллектуальных систем представляли собой экспертные системы, ориентированные на конкретные предметные области. Однако победа высокоуровне-

вой стратегии оказалась временной. Конец XX в.—начало XXI в. ознаменовались широкомасштабным наступлением нейросетевых и нейрокомпьютерных технологий. Причем имело место как применение этих технологий в новых предметных областях, так и вытеснение экспертных систем из тех областей, где они традиционно применялись ранее.

Уже в течение полувека между высоко- и низкоуровневой стратегиями создания интеллектуальных систем идет жесточайшая конкуренция. Вместе с тем многие специалисты отмечают наметившийся в последнее время процесс интеграции конкурирующих стратегий. Известны, например, успешные попытки создания и применения *гибридных интеллектуальных систем*, комбинирующих идеи обеих этих альтернативных стратегий.

Третья стратегия, появившаяся на свет в середине 1970-х гг. благодаря работам профессора Мичиганского университета Дж. Холланда, навеяна Дарвиновской теорией возникновения и эволюции жизни на Земле. *Эволюционные модели* начинают свою работу с создания целой популяции особей — кандидатов на решение проблемы. Отдельные особи популяции оцениваются по определенному критерию, позволяющему отобрать лучшие, которые затем видоизменяются, наследуя положительные свойства родителей и образуя новые поколения. Этот подход предполагает поиск источника интеллекта в самом процессе эволюции и взаимодействия особей. Знания в эволюционных моделях приобретаются в ходе конкурентной борьбы между особями.

Подводя итог краткого обзора существующих стратегий создания интеллектуальных систем, отметим, что на сегодняшний день безусловным лидером как в области теоретических исследований, так и в части практических приложений являются нейросетевые технологии. Моделирование мозга на самом низком уровне абстрагирования — на уровне нейронной структуры — приводит к наиболее плодотворным технологиям создания интеллектуальных информационных систем.

1.3. ЛИДИРУЮЩАЯ НАУЧНАЯ ОТРАСЛЬ

В разные периоды существования цивилизации лидирующими направлениями научных исследований были философия, математика, астрология, алхимия (средние века), разделы физики: механика (XIII в.), электричество и магнетизм (XIX в.), квантовая механика, физика элементарных частиц, ядерная физика (конец XIX в.—первая половина XX в). Со второй половины XX в. и по настоящее время лидерство, как в темпах развития, так и в приложениях, перешло к *информатике* (рис. 1.4). В свою очередь, среди множества разделов информатики наиболее экспансирующим, как в области теоретических разработок, так и приложениях, является *искусственный интеллект*.

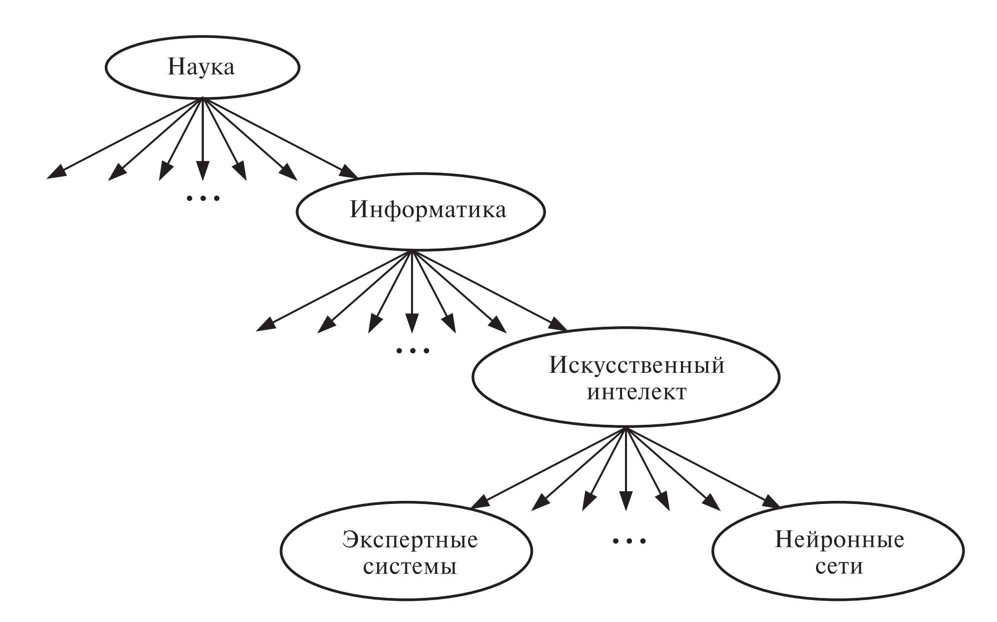


Рис. 1.4. Современное распределение приоритетных направлений развития научных отраслей

Искусственный интеллект также имеет несколько разделов, причем приоритеты между этими разделами постоянно меняются. Более того, между учеными, развивающими различные разделы искусственного интеллекта, происходит конкуренция, зачастую принимающая весьма жесткие формы. И победителями в этой конкуренции попеременно оказываются то одни, то другие: ажиотаж 1950-х гг. вокруг первого нейрокомпьютера сменяется лидерством *экспертных систем*, которое продолжается до середины 1980-х гг. Однако в течение двух последних десятилетий мы наблюдаем, как экспертные системы постепенно уступают лидерство нейронным сетям.

Искусственный интеллект уже давно стал «горячей точкой» научных исследований. Здесь, как в фокусе, сконцентрированы усилия математиков, физиков, нейробиологов, психологов, программистов, философов, инженеров. Здесь решаются глобальные вопросы, связанные с путями развития научной мысли, с воздействием компьютерной индустрии на жизнь будущих поколений. Здесь рождаются и получают право на жизнь пограничные идеи — результат объединения различных дисциплин, научных направлений и областей. Здесь возникает и формируется то, что называется философским осмыслением научных результатов.

Быстрому развитию искусственного интеллекта как научной отрасли в значительной мере способствует ставшая повсеместной доступность компьютерной техники. По словам известного французского матема-

тика Ж. А. Пуанкаре, «наука — это кладбище гипотез». Любая догадка, любая новая идея сначала является научной гипотезой и должна быть либо подтверждена, либо отвергнута в ходе экспериментальной проверки. В иных научных областях на проверку гипотез уходят десятилетия. Но это не относится к современному искусственному интеллекту. Благодаря доступности компьютеров существует возможность быстрого компьютерного воплощения новых идей, возможность их немедленной реализации и всесторонней практической проверки. Такая возможность является сильнейшим стимулом, способствующим лидерству искусственного интеллекта в темпах его развития, как в теоретическом, так и в практическом аспектах.

1.4. ОСНОВНЫЕ СФЕРЫ ПРИМЕНЕНИЯ

Сегодня искусственный интеллект — это обширная область исследований и разработок интеллектуальных систем, предназначенных для самых разнообразных сфер человеческой деятельности. Для задач, решаемых методами искусственного интеллекта, характерно наличие большого числа степеней свободы с числом вариантов поиска решений, приближающимся к бесконечности. В отличие от жестко детерминированных компьютерных программ системы искусственного интеллекта сами ищут пути решения поставленной задачи. При этом они могут менять свои параметры и структуру, чутко реагировать на изменения внешней среды, самообучаться, самосовершенствоваться и развиваться, жить самостоятельной, не зависящей от разработчика жизнью.

Распознавание образов. К распознаванию образов в искусственном интеллекте относят широкий круг проблем: распознавание изображений, символов, текстов, запахов, звуков, шумов, ситуаций. На рынке программных средств имеются системы, основанные на распознавании по признакам, оснащенные базами данных и знаний, имеющие возможность адаптации и обучения. Однако в последнее время становятся популярными гибридные системы, в которых наряду с технологиями экспертных систем используются и нейросетевые технологии.

Игры. Традиционно искусственный интеллект включает интеллектуальные задачи, решаемые при игре в шахматы, шашки, го, каллах. Здесь очень широко применяется один из методов — лабиринтная модель плюс эвристики. Кроме того, в современных программах-игроках наиболее полно удалось реализовать центральную идею искусственного интеллекта — обучение, самообучение и самоорганизацию.

В широком смысле слова под *игрой* понимается некая конфликтная ситуация, участники которой своими действиями не только достигают своих личных целей, но и влияют на достижимость целей другими участниками игры. Ясно, что под такое толкование игры подпадают многие экономические, политические и военные конфликты.

Компьютерное творчество. Методы искусственного интеллекта применяются для создании компьютерных программ, моделирующих творческую деятельность человека. Пока только теоретический интерес представляют программы, сочиняющие музыкальные и поэтические произведения, создающие художественные полотна. Практический интерес представляет моделирование творческой деятельности ученых, например математиков, доказывающих теоремы, применяющих нестандартные методы решения краевых задач. Так, в работе [12] дано описание программы REGIONS, предназначенной для нахождения аналитических решений краевых задач математической физики. Подобные системы обладают базами знаний, содержащими нужные теоремы, математические зависимости и эвристические правила, обобщающие опыт и интуицию математика, способны к обучению с помощью учителя и к самообучению. Интерес к «компьютерным математикам» обусловлен главным образом тем, что они способны получать точные аналитические решения краевых задач [68, 71, 72, 77, 80–81, 85–87]. Такие решения не имеют вычислительных погрешностей, что весьма актуально для инженерных расчетов экологически опасных объектов и процессов ответственного назначения.

Компьютерная лингвистика. Начиная с 50-х гг. XX в. и по настоящее время одной из популярных тем исследований искусственного интеллекта является область машинного перевода текстов с одного языка на другой. В 1954 г. в США была создана компьютерная программа, которая успешно перевела с одного языка на другой 60 фраз. В СССР работы в области машинного перевода начались в 1955 г. с создания программы-переводчика с английского языка на русский.

В первых программах-переводчиках применялась идея пословного перевода, что, естественно, не позволяло получать качественные результаты из-за большого количества синонимов и многозначных слов естественных языков. Постепенно структуры естественно-языковых интерфейсов усложнялись, и в настоящее время в программах-переводчиках в обязательном порядке применяются:

- морфологический анализ — анализ слов в тексте;
- синтаксический анализ — анализ предложений, грамматики и связей между словами;
- семантический анализ — анализ смысла каждого предложения на основе базы знаний, на которую ориентирована конкретная программа-переводчик;
- прагматический анализ — анализ смысла предложений в окружающем контексте с помощью базы знаний.

Добавление баз знаний и использующих их анализаторов в программах-переводчиках позволило существенно улучшить качество переводов. Особенно ощутимых результатов удалось добиться в переводах узкоспециализированных текстов, таких как текст прогноза погоды.

Тем не менее компьютерные переводы, как правило, весьма далеки от совершенства, но, несмотря на это, они широко применяются для обработки технических, коммерческих, правительственных документов, а также документов, опубликованных в Интернете.

Другой проблемой компьютерной лингвистики является разработка естественно-языкового интерфейса между человеком и машиной. Здесь немаловажную роль могут сыграть нейросетевые технологии, с помощью которых удается научить компьютер правильному произношению слов. В проектах создания компьютеров V-го и VI-го поколений решению этой проблемы уделено первостепенное внимание.

Интеллектуальные роботы. Роботы — это технические устройства, предназначенные для автоматизации человеческого труда. Само слово «робот» появилось в 20-х гг. XX в. Его автор — чешский писатель Карел Чапек.

В настоящее время в промышленности применяется огромное количество роботов-манипуляторов, работающих по жесткой схеме управления. В отличие от них интеллектуальные роботы обладают способностью самообучаться и самоорганизовываться, адаптироваться к изменяющейся окружающей обстановке, принимать самостоятельные решения.

При создании первых интеллектуальных роботов в них закладывались базы правил, предписывающих их поведение. Например, закладывалась инструкция: «Поднять правую ногу на 15 см, вытянуть ее вперед на 20 см и поставить на ступеньку, опустив вниз на 30 см», и, если этой ступеньки не оказывалось, робот падал. Затем для управления поведением роботов стали применяться нейрокомпьютерные технологии. Благодаря этому роботы приобрели способность адаптироваться к окружающей среде, гибко реагировать на ее изменения, самонастраиваться и самообучаться.

Интересные эксперименты проводятся по применению эволюционных моделей управления роботами. Снабжение их природными механизмами самосовершенствования живых организмов, таких как селекция, скрещивание, мутация, естественный отбор, представляется весьма перспективным.

Интерес, а иногда и недоумение вызывают проекты, связанные с созданием антропоморфных роботов — максимально имитирующих человека. В начале XXI в. лидером капиталовложений в проекты создания антропоморфных роботов стала Япония. Достижениями японского проекта стал робот, играющий на флейте, как человек, передвигающийся на двух ногах, как человек, улыбающийся и кланяющийся, как человек и т. д.

Россия в настоящее время имеет приоритет в области разработки математической теории эмоциональных роботов благодаря пионерским монографиям профессора О. Г. Пенского с соавторами [42, 43].

Искусственная жизнь, генетические алгоритмы, мультиагентные системы. Эти три направления берут свое начало с опытов

профессора Мичиганского университета Дж. Холланда, который, пытаясь подтвердить Дарвиновскую теорию эволюции жизни на Земле, населил компьютер виртуальными особями-программами, заставив их, подобно реальным земным существам, бороться за выживание, добывать себе пищу, размножаться и умирать. Научными работами подобного рода под общим названием «Искусственная жизнь» и сейчас занимаются многие биологи, экологи, ученые, работающие в области наук о Земле и человеке. Однако широкое практическое приложение идей Дж. Холланда началось, когда было обнаружено, что дарвиновский алгоритм самосовершенствования всего живого может с успехом применяться для решения оптимизационных задач. Выяснилось, что природные оптимизационные алгоритм, названные *генетическими*, легко справляются с поисками глобальных экстремумов многоэкстремальных функций, с чем традиционные градиентные методы, как правило, справиться не могут. В настоящее время генетические алгоритмы активно применяются при решении многих оптимизационных задачах, встречающихся преимущественно в экономико-математическом моделировании, а также при обучении нейронных сетей.

Другое практическое применение алгоритмов искусственной жизни связано с использованием *агентных* (*мультиагентных*) *систем*. Программами-особями, называемыми агентами, населяется сеть Интернет. Действуя самостоятельно, агенты выполняют полезную работу в интересах своего резидента — пользователя. Они следят за событиями в мире, добывая полезную информацию, играют на бирже ценных бумаг, добиваясь максимальных дивидендов для хозяина, сообщают ему об угрозах и предотвращают их, иногда вступая в борьбу с чужими агентами.

Компьютерные вирусы. Сегодня трудно назвать компьютерного пользователя, избежавшего знакомства с этим видом программной продукции.

Следует признать, что последние поколения вирусов обладают всеми свойствами систем искусственного интеллекта. Они свободно перемещаются по компьютерам, мутируют и размножаются, самообучаются, самостоятельно изменяют свои параметры и структуру. Вирусы живут самостоятельной, независимой от их создателей жизнью.

Вредоносное воздействие компьютерных вирусов значительно возросло с появлением сети Интернет. Однако, по прогнозам специалистов, неприятности, с которыми мы встречаемся сегодня, представляются ничтожными по сравнению с теми, которые ожидают нас с проникновением компьютерных вирусов в сферу деятельности интеллектуальных роботов.

Интеллектуальный анализ данных (*Data mining*) — извлечение знаний из данных. Эта задача стала особенно актуальной в связи с обилием статистической информации, которой буквально «забита»

сеть Интернет и компьютеры, обслуживающие деятельность различного рода фирм и организаций. Однако за множеством безликих цифр скрываются закономерности. Эти закономерности представляют собой знания, которые необходимы менеджерам для выработки верных решений. Но увидеть такие закономерности человеческий глаз не в состоянии. Незаменимым инструментом создания систем извлечения знаний из данных и поддержки принятия решений на их основе являются методы искусственного интеллекта, в частности — нейросетевые технологии.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Попытайтесь самостоятельно сформулировать определения интеллекта и искусственного интеллекта.
2. Когда и кем была произведена первая попытка моделирования интеллекта человека?
3. Опишите назначение и принцип действия машины Р. Луллия.
4. В чем суть модели лабиринтного поиска и эвристического метода?
5. В чем состоит принципиальное различие между кибернетикой «черного ящика» и нейрокибернетикой? В чем смысл терминов «восходящее» и «нисходящее» направления искусственного интеллекта? Что такое эволюционное моделирование?
6. Перечислите и охарактеризуйте основные стратегии создания интеллектуальных систем. Назовите основателя и примерную дату появления каждого стратегического направления.
7. Как приобретаются и в каком виде хранятся знания в интеллектуальных системах, созданных согласно трем основным стратегиям?
8. Какая научная область, по вашему мнению, является лидирующей в настоящее время?
9. Зайдите в сеть Интернет и с помощью поисковика попробуйте ввести ключевые слова: «искусственный интеллект», «экспертные системы», «нейронные сети», «нейроинформатика», «нейрокомпьютер», «агентные системы», «интеллектуальные роботы», «интеллектуальный анализ данных». Выберите заинтересовавшую вас информацию и напишите на ее основе реферат или подготовьте доклад на какую-либо из следующих тем, например:
   - успехи искусственного интеллекта;
   - история искусственного интеллекта;
   - современные проблемы искусственного интеллекта;
   - современные технологии создания интеллектуальных систем;
   - будущее искусственного интеллекта.

Глава 2

МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

2.1. ДАННЫЕ И ЗНАНИЯ

Как правило, любая, даже самая простая, компьютерная программа оперирует не только с данными, но и со знаниями. Например, фрагмент программы, предназначенной для вычисления площади круга радиусом 20 см, на языке Pascal выглядит следующим образом:

```
%
R := 20;
Pi := 3.14;
S := Pi * R * R;
WRITELN ('Площадь круга S=', S, 'кв. см');
```

Первые два оператора представляют собой данные, а третий оператор — знание. Это всем известная формула для вычисления площади круга. Она является результатом интеллектуальной деятельности великих геометров древности.

Прежде чем сформулировать определение знаний, вспомним [5, 23], что собой представляют данные.

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства.

При обработке данные последовательно трансформируются:

- данные, существующие как результат измерений и наблюдений;
- данные на материальных носителях информации — в таблицах, протоколах, справочниках;
- данные, представленные в виде диаграмм, графиков, функций;
- данные в компьютере на языке описания данных;
- базы данных.

Знания основываются на данных, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе практической деятельности или научных исследований. Они могут выражать законы природы и общества, закономерности конкретных предметных областей.

При обработке на ЭВМ знания трансформируются аналогично данным:

- знания, существующие в памяти человека как результат обучения, воспитания, мышления;

- знания, помещенные на материальных носителях: в учебниках, инструкциях, методических пособиях, книгах;
- знания, описанные на языках представления знаний и помещенные в компьютер;
- базы знаний.

Для хранения данных в компьютере используются базы данных. Для них характерны большой объем и относительно небольшая стоимость информации. Для хранения знаний используются базы знаний. Они, наоборот, отличаются сравнительно небольшими объемами, но исключительно дорогими информационными массивами.

В приведенном выше фрагменте программы вычисления площади круга знания растворены в самом тексте программы. Такой вид представления знаний называют *процедурным*. Корректировка таких знаний требует изменения самого текста программы. Поэтому с развитием искусственного интеллекта все большая часть знаний стала сосредоточиваться в отдельных структурах. Такие знания называются *декларативными*.

Существуют десятки способов представления знаний [23, 25, 40]. Мы кратко рассмотрим только пять наиболее известных способов:

- продукционная модель;
- фреймовая модель;
- сетевая модель;
- логическая модель;
- синаптическая модель.

2.2. ПРОДУКЦИОННАЯ МОДЕЛЬ

Продукционная система состоит из трех основных компонентов, схематично изображенных на рис. 2.1. Первый из них — это база правил, состоящих из правил типа: ЕСЛИ (условие), ТО (действие): ЕСЛИ холодно, ТО надеть шубу; ЕСЛИ идет дождь, ТО взять зонтик и т. п.

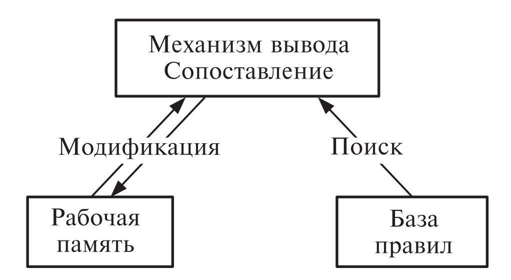


Рис. 2.1. Блок-схема продукционной системы

Вторым компонентом является рабочая память, в которой хранятся исходные данные к задаче и выводы, полученные в ходе работы системы.

Третий компонент — механизм логического вывода, использующий правила в соответствии с содержимым рабочей памяти.

Рассмотрим конкретный пример [45, 46]. В базе правил экспертной системы имеются два правила.

***Правило* 1:** ЕСЛИ «намерение — отдых» и «дорога ухабистая», ТО «использовать джип».

***Правило* 2:** ЕСЛИ «место отдыха — горы», ТО «дорога ухабистая».

Допустим, что в рабочую память поступили исходные данные:

«намерения — отдых»; «место отдыха — горы».

Механизм вывода начинает сопоставлять образцы из условных частей правил с образцами, хранимыми в рабочей памяти. Если образцы из условной части имеются в рабочей памяти, то условная часть считается истинной. В противном случае — ложной.

В данном примере при рассмотрении правила 1 оказывается, что образец «намерение — отдых» имеется в рабочей памяти, а образец «дорога ухабистая» отсутствует, поэтому условная часть правила 1 считается ложной. При рассмотрении правила 2 выясняется, что его условная часть истинна. Механизм вывода выполняет заключительную часть этого правила, и образец «дорога ухабистая» заносится в рабочую память. Правило 2 при этом выбывает из числа кандидатов на рассмотрение.

Снова рассматривается правило 1, условная часть которого теперь становится истинной, и содержимое рабочей памяти пополняется образцом «использовать джип». В итоге получается, что правил, которые можно было бы применять, не остается и система останавливается.

В рассмотренном примере приведен прямой вывод — от данных к поиску цели. Однако применяют и обратный вывод — от цели для ее подтверждения к данным. Продемонстрируем этот способ на нашем примере. Допустим, что наряду с исходными данными «намерения — отдых»; «место отдыха — горы» имеется цель «использовать джип».

Согласно правилу 1, для достижения этой цели требуется выполнение условия «дорога ухабистая», поэтому условие становится новой целью. При рассмотрении правила 2 оказывается, что условная часть этого правила в данный момент истинна, поэтому рабочая память пополняется образцом «дорога ухабистая». При повторном рассмотрении правила 1 подтверждается цель: «использовать джип».

При обратном выводе система останавливается в двух случаях: либо достигается первоначальная цель, либо кончаются правила. При прямом выводе система останавливается только тогда, когда кончаются правила, либо при появлении в рабочей памяти специально предусмотренного образца, например «использовать джип».

В приведенном примере на каждом этапе прямого вывода можно было использовать только одно правило. В общем же случае на каж-

дом этапе вывода таких правил несколько, и тут возникает проблема выбора. Например, введем в рассмотрение еще одно правило.

***Правило* 3:** ЕСЛИ «намерение — отдых», ТО «нужна скорость».

Кроме того, введем условие останова системы — появление в рабочей памяти образца «использовать джип».

Теперь на первом этапе прямого вывода появляется возможность применять либо правило 2, либо правило 3. Если сначала применить правило 2, то на следующем этапе можно будет применять правило 1 и правило 3. Если на этом этапе применить правило 1, то выполнится условие останова системы, но если прежде применить правило 3, то потребуется еще один этап вывода.

Этот пример показывает, что выбор применяемого правила оказывает прямое влияние на эффективность вывода. В реальной системе, где имеется множество правил, появляется проблема их оптимального выбора.

Если на каждом этапе логического вывода существует множество применимых правил, то это множество носит название *конфликтного набора*, а выбор одного из них называется *разрешением конфликта*.

Аналогичная ситуация возникает и при обратном выводе. Например, дополним предыдущий пример еще одним правилом.

***Правило* 4:** ЕСЛИ «место отдыха — пляж», ТО «дорога ухабистая».

Если на основании этого условия подтверждается цель «использовать джип», то для достижения первоначальной цели достаточно применить только одно правило 1; однако, чтобы подтвердить новую цель «дорога ухабистая», открывается возможность применения правила 1, нужно использовать либо правило 2, либо правило 4. Если сначала применить правило 2, то это будет самый удачный выбор, поскольку сразу же можно применить и правило 1. В то же время, если попытаться применить правило 2, то, поскольку образца «место отдыха — пляж», который является условием правила 4, в рабочей памяти не существует и, кроме того, не существует правила, подтверждающего его, данный выбор является неудачным. И лишь со второго захода, применяя правило 2, можно подтвердить цель «дорога ухабистая».

Следует обратить внимание на то, что при обратном выводе правило 3, которое не оказывает прямого влияния на достижение цели, не принималось в расчет с самого начала. Таким образом, для обратных выводов характерна тенденция исключения из рассмотрения правил, не имеющих прямого отношения к заданной цели, что позволяет повысить эффективность вывода. Обратный вывод применяют в тех случаях, когда цели известны и их сравнительно немного.

Продукционная модель — это наиболее часто используемый способ представления знаний в современных экспертных системах. Основными преимуществами продукционной модели являются наглядность, высокая модульность, легкость внесения изменений и дополнений, простота механизма логического вывода.

2.3. ФРЕЙМОВАЯ МОДЕЛЬ

В психологии и философии используется понятие абстрактного образа. Например, слово «автомобиль» вызывает у слушающих образ устройства, способного перемещаться, имеющего четыре колеса, салон для шофера и пассажиров, двигатель, руль. Считается, что современный человек использует абстрактные образы для хранения в своей памяти информации об окружающем мире.

Фрейм — это модель абстрактного образа, которую программисты используют для хранения знаний о рассматриваемой предметной области. Фрейм состоит из имени и отдельных единиц, называемых ***слотами***. Он имеет однородную структуру:

Имя фрейма
Имя 1-го слота: значение 1-го слота.
Имя 2-го слота: значение 2-го слота.
. . . .
Имя N-го слота: значение N-го слота.

В качестве значения слота может выступать имя другого фрейма. Таким образом фреймы объединяются в сеть. Свойства фреймов наследуются сверху вниз, т. е. от вышестоящих к нижестоящим через АКО-связи (начальные буквы английских слов «A Kind Of», что можно перевести как «это»). Слот с именем АКО указывает на имя фрейма более высокого уровня иерархии.

Например, на рис. 2.2 фрейм «Студент» имеет ссылки на вышестоящие фреймы: «Человек» и «Млекопитающее». Поэтому на вопрос «Может ли студент мыслить?» ответ будет положительным, так как этим свойством обладает вышестоящий фрейм «Человек».

Если одно и то же свойство указывается в нескольких связанных между собой фреймах, то приоритет отдается нижестоящему фрей-

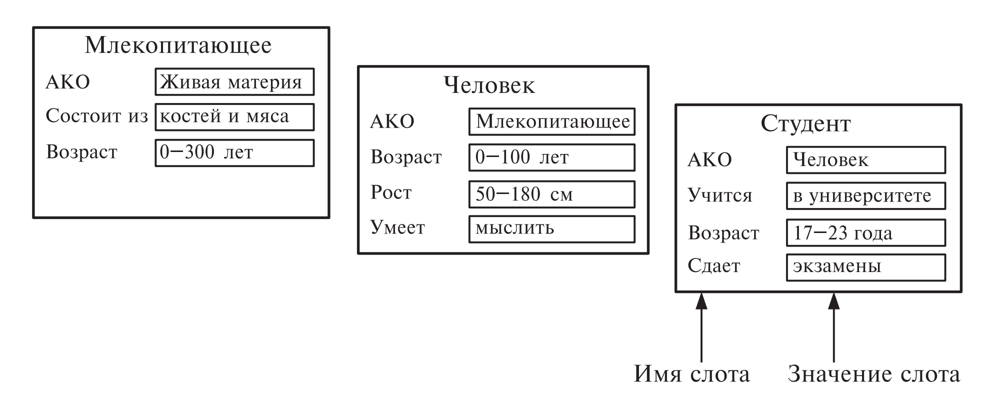


Рис. 2.2. Структура фреймов

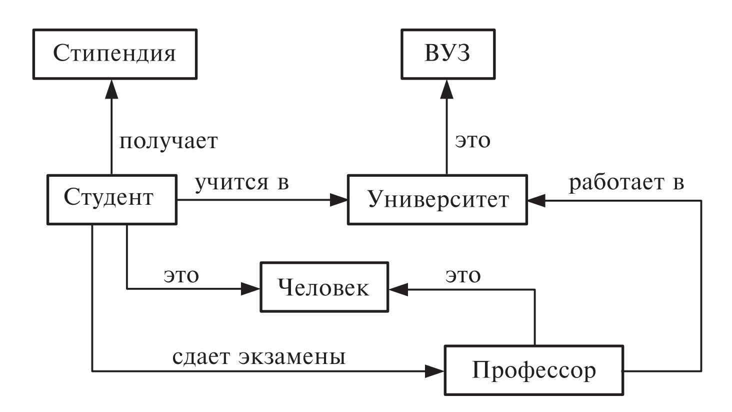


Рис. 2.3. Семантическая сеть

му. Так, возраст фрейма «Студент» не наследуется из вышестоящих фреймов.

Основным преимуществом фреймов как способа представления знаний является наглядность и гибкость в употреблении. Кроме того, фреймовая структура согласуется с современными представлениями о хранении информации в памяти человека.

2.4. СЕТЕВАЯ МОДЕЛЬ

В основе этого способа представления знаний лежит идея о том, что любые знания можно представить в виде совокупности *понятий* (объектов) и *отношений* (связей). На рисунке 2.3. приведен пример графического изображения сети, вершины которой представляют собой понятия предметной области, а связывающие их линии — отношения между этими понятиями. Сам термин «*семантическая*» означает «смысловая».

Основным преимуществом этой модели является наглядность, а также соответствие современным представлениям об организации долговременной памяти человека. Недостаток — сложность поиска вывода, а также сложность корректировки, т. е. удаления и дополнения сети новыми знаниями.

2.5. ЛОГИЧЕСКАЯ МОДЕЛЬ

Логическая модель знаний представляет собой совокупность утверждений. О каждом утверждении можно сказать, истинно оно или ложно. Утверждения делятся на факты и правила. Совокупность фактов представляет собой базу данных, лежащую в основе базы знаний. Правила имеют форму «ЕСЛИ А, ТО Б». Механизм вывода основан

на аппарате математической логики, называемом исчислением предикатов первого порядка. Логическая модель знаний лежит в основе языка ПРОЛОГ.

Прикладные возможности этой модели весьма ограничены.

2.6. СИНАПТИЧЕСКАЯ МОДЕЛЬ

Идея этой модели заимствована у биологов, представляющих мозг в виде нервных клеток — *нейронов*, соединенных нервными волокнами, и обменивающихся между собой электрическими сигналами, которые передаются по нервным волокнам. Согласно этим представлениям, вся информация и все знания, которыми располагает мозг, закодированы в виде матрицы *сил синаптических связей*, под которыми понимаются электропроводности *синапсов* — точек контактов *нервных волокон* с *дендритами* — отростками нейронов, принимающих электрические сигналы.

Попытки компьютерного моделирования электроинформационных процессов, происходящих в мозге, оказались чрезвычайно плодотворными и привели к появлению нового раздела искусственного интеллекта — нейросетевым и нейрокомпьютерным технологиям, подробному изучению которых посвящена гл. 4.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите общие и отличительные признаки данных и знаний.
2. Назовите и охарактеризуйте известные вам методы представления знаний.
3. Какой, по вашему мнению, метод представления знаний используется в человеческом мозге?

Глава 3

ЭКСПЕРТНЫЕ СИСТЕМЫ

3.1. ПРЕДМЕТНЫЕ ОБЛАСТИ

Знания, которыми обладает специалист в какой-либо области, можно разделить на формализуемые, плохо формализуемые и не формализуемые. Формализуемые знания излагаются в книгах и руководствах в виде законов, формул, моделей, алгоритмов. Формализуемые знания характерны для точных наук, таких как математика, физика, химия, астрономия. Науки, которые принято называть описательными, обычно оперируют с плохо формализуемыми знаниями. К таким наукам можно отнести, например, зоологию, ботанику, экологию, педагогику, медицину и др.

Существуют знания, которые не попадают в книги в связи с их неконкретностью, субъективностью, приблизительностью. Знания этого рода являются результатом многолетних наблюдений, опыта работы, интуиции. Они обычно представляют собой некие эмпирические и эвристические приемы и правила. Обычно они передаются из поколения в поколение в виде определенных навыков, ноу-хау, секретов ремесла.

Есть также знания, которые не могут быть выражены ни в математическом виде, ни в терминах обычного человеческого языка. Такими знаниями обладают религиозные деятели, экстрасенсы, контактеры, шаманы.

Класс задач, относящихся к плохо формализуемым знаниям и не формализуемым, значительно шире класса задач, для которых знания могут быть легко формализованы. Этим объясняется особая популярность и широкое практическое применение экспертных систем, которые сделали возможным применение компьютерных технологий в предметных областях, в которых знания плохо поддаются формализации.

3.2. СТРУКТУРА И РЕЖИМЫ РАБОТЫ

Экспертные системы — это сложные программные комплексы, аккумулирующие знания специалистов в конкретных предметных областях и использующие эти знания с целью выработки логически обоснованных рекомендаций и решений проблем, а также для консультаций менее квалифицированных пользователей [23]. Типичная структура экспертной системы изображена на рис. 3.1.

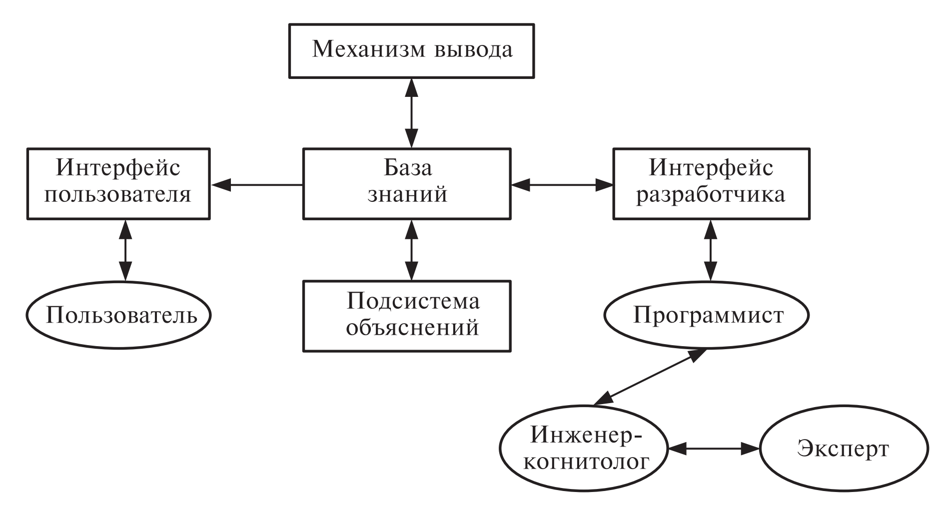


Рис. 3.1. Типичная структура экспертной системы

Обычно в состав экспертной системы входят следующие взаимосвязанные между собой модули.

База знаний – ядро экспертной системы, содержит знания, представленные одним из способов представления знаний, рассмотренных выше.

Интерфейс разработчика — программа, с помощью которой инженер-когнитолог и программист могут создавать базу знаний в диалоговом режиме. Включает в себя системы вложенных меню, шаблонов языка представления знаний, подсказок (help-режим) и других сервисных средств, облегчающих работу с базой знаний.

Интерфейс пользователя — комплекс программ, реализующих диалог пользователя с экспертной системой на стадии как ввода информации, так и получения результатов.

Механизм вывода (синонимы: *дедуктивная машина, блок логического вывода, решатель*) — программа, моделирующая ход рассуждений эксперта на основании знаний, имеющихся в базе знаний. Механизм вывода определяет, какие правила (знания) нужно вызвать, выполняет правила, определяет, когда найдено приемлемое решение, передает результат программе «*Интерфейс пользователя*».

Подсистема объяснений — программа, позволяющая пользователю получать ответы на вопросы: «Как была получена та или иная рекомендация?» и «Почему система приняла такое решение?». Ответ на вопрос «Как?» — это трассировка (последовательный вывод на монитор или в печать) всего процесса получения решения с указанием исполняющих фрагментов базы знаний, т. е. всех шагов цепи умозаключений. Ответ на вопрос «Почему?» — ссылка на умозаключение, непосредственно предшествовавшее полученному решению, т. е. отход на один шаг назад.

В коллектив разработчиков экспертной системы входят как минимум четыре специалиста (или четыре группы специалистов): эксперт, инженер-когнитолог, программист, пользователь. Возглавляет коллектив инженер-когнитолог — ключевая фигура при разработке систем, основанных на знаниях. Обычно это руководитель проекта, в задачу которого входит организация всего процесса создания экспертной системы. С одной стороны, он должен быть специалистом в области искусственного интеллекта, а с другой — должен разбираться в предметной области, общаться с экспертом, извлекая и формализуя его знания, передавать их программисту, который кодирует и помещает знания в базу знаний экспертной системы.

Экспертная система работает в двух режимах: приобретения знаний и решения задач или консультаций.

В режиме *приобретения знаний* происходит формирование базы знаний. В режиме *решения задач* общение с экспертной системой осуществляет конечный пользователь.

Обычно знания, которыми располагает эксперт, различаются степенью надежности, важности, четкости. В этом случае они снабжаются некоторыми весовыми коэффициентами, которые называют *коэффициентами доверия* (*уверенности*). Такие знания обрабатываются с помощью алгоритмов нечеткой логики.

В процессе опытной эксплуатации коэффициенты доверия могут подвергаться корректировке. В этом случае говорят, что происходит обучение экспертной системы. Процесс обучения экспертной системы может производиться автоматически с помощью обучающего алгоритма либо путем вмешательства инженера-когнитолога, выполняющего роль учителя.

Если в процессе работы экспертной системы механизм вывода обнаруживает, что можно вызвать более одного правила, то делается выбор, причем приоритет отдается либо правилам, имеющим более высокий коэффициент доверия, либо правилам, которые учитывают большее количество текущих данных. Этот процесс называется *разрешением конфликта*.

3.3. ЭТАПЫ И ТЕХНОЛОГИЯ РАЗРАБОТКИ

Технология разработки экспертных систем несколько отличается от технологии разработки обычных компьютерных программ. Это различие обусловлено разницей их характеристик (табл. 3.1).

При разработке экспертных систем широко используются *прототипы* — компьютерные модели будущего программного продукта. Идея прототипизации состоит в том, чтобы на ранней стадии работы над проектом создать упрощенную версию конечной программы, которая могла бы послужить доказательством продуктивности основных идей,

Таблица 3.1

Характеристика	**Экспертные системы**	**Традиционные компьютерные программы**
Тип обработки	Символьная	Числовая
Метод	Эвристический поиск	Алгоритм
Задание шагов решения	Неявное	Точное
Искомое решение	Удовлетворительное	Оптимальное
Управление и данные	Перемешаны	Разделены
Знания	Неточные	Точные
Модификации	Частые	Редкие

положенных в основание проекта. Прототип должен быть способен решать какую-либо из нетривиальных задач, характерных для заданной области применения. На основе анализа опыта работы с прототипом разработчики могут уточнить требования к системе в целом и ее основным функциональным характеристикам. Заказчики экспертной системы получают возможность оценить целесообразность выполнения проекта еще до того, как на него будут потрачены значительные средства.

Принято различать следующие прототипы экспертных систем [23].

Демонстрационный прототип — экспертная система, которая решает часть требуемых задач, демонстрируя жизнеспособность метода инженерии знаний. База знаний составляет всего 50…100 правил. Время разработки такой экспертной системы — 6…12 мес.

Исследовательский прототип — экспертная система, которая решает все требуемые задачи, но неустойчива в работе и не полностью проверена. База знаний содержит 200…500 правил. Разработка занимает 3…6 мес.

Действующий прототип — надежно решает все задачи, но для решения сложных задач может потребоваться много времени и памяти. База знаний содержит 500…1000 правил. Время разработки — 6…12 мес.

Промышленная экспертная система — обеспечивает высокое качество решения всех задач при минимуме времени и памяти, что достигается переписыванием программ с использованием более совершенных инструментальных средств и языков низкого уровня. База знаний содержит 1000…1500 правил. Время разработки — 1…1,5 года.

Коммерческая экспертная система — отличается от промышленной тем, что, помимо собственного использования, она может продаваться различным потребителям. База знаний содержит 1500…3000 правил. Время разработки — 1,5…3 года. Стоимость — 0,3…5,0 млн долл.

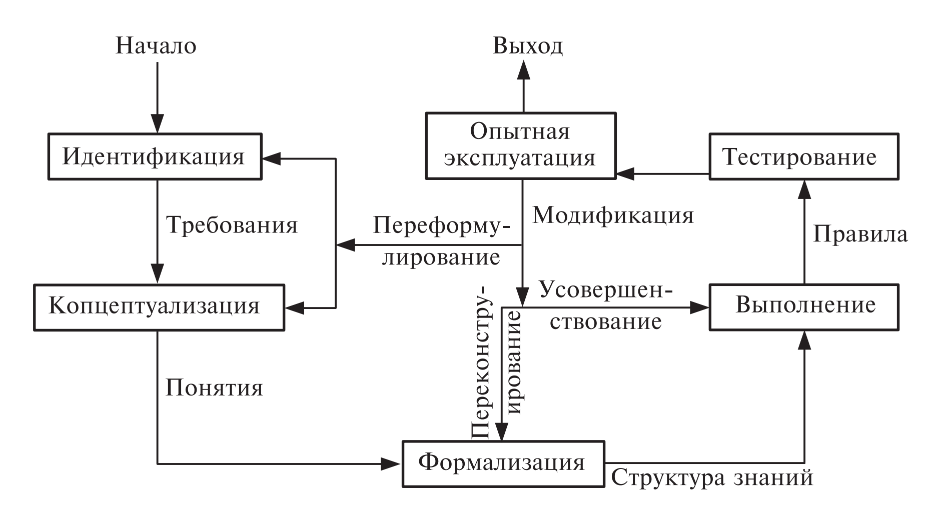


Рис. 3.2. Технология разработки экспертной системы

В ходе работ по созданию экспертных систем сложилась определенная технология, включающая основные этапы (схематически они изображены на рис. 3.2).

1. *Идентификация* (постановка задачи). На этом этапе устанавливаются задачи, которые подлежат решению, выявляются цели разработки, требования к экспертной системе, ресурсы, используемые понятия и их взаимосвязи, определяются методы решения задач. Цель этапа — сформулировать задачу, охарактеризовать поддерживающую ее базу знаний и таким образом обеспечить начальный импульс для развития базы знаний.

2. *Концептуализация*. Проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

3. *Формализация*. Определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, оценивается, насколько адекватны целям системы зафиксированные понятия, методы решения, средства представления и манипулирования знаниями.

4. *Выполнение*. На этом этапе эксперт наполняет базу знаний. Процесс приобретения знаний разделяют на получение знаний от эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном экспертной системе. Из-за эвристического характера знаний их приобретение является весьма трудоемким.

5. *Тестирование*. Эксперт и инженер по знаниям в интерактивном режиме, используя диалоговые и объяснительные средства, проверяют

компетентность экспертной системы. Процесс тестирования продолжается до тех пор, пока эксперт не решит, что система достигла требуемого уровня компетентности.

6. *Опытная эксплуатация*. Проверяется пригодность экспертной системы для конечных пользователей. По результатам этого этапа может потребоваться модификация экспертной системы.

7. *Модификация*. В ходе создания экспертной системы почти постоянно производится ее модификация: переформулирование понятий и требований, переконструирование представления знаний и усовершенствование прототипа.

Усовершенствование прототипа осуществляется в процессе циклического прохождения через этапы выполнения и тестирования для отладки правил и процедур вывода.

Переконструирование выбранного ранее способа представления знаний предполагает возврат от этапа тестирования на этап формализации.

Если возникшие проблемы еще более серьезны, то после неудачи на этапе тестирования может потребоваться возврат на этап концептуализации и идентификации. В этом случае речь идет о переформулировании понятий, используемых в системе, т. е. перепроектировании системы заново.

Приведенная последовательность разработки экспертных систем предложена Э. А. Поповым в книге [44]. Данная технология отражает опыт разработки и внедрения многочисленных экспертных систем широкого назначения. В этой же книге указываются трудности, характерные для каждой стадии, и даются рекомендации по их преодолению.

3.4. ПРОГРАММНЫЙ ИНСТРУМЕНТАРИЙ

Инструментальные программы, применяемые при разработке экспертных систем, можно разделить на четыре достаточно большие категории.

1. Оболочки экспертных систем. Системы этого типа создаются, как правило, на основе какой-нибудь экспертной системы, достаточно хорошо зарекомендовавшей себя на практике. При создании оболочки из экспертной системы-прототипа удаляются компоненты, слишком специфичные для области ее непосредственного применения, и оставляются те, которые не имеют узкой специализации. Примером может служить система EMYCIN, созданная на основе широко известной экспертной системы медицинской диагностики MYCIN.

Достоинством программ-оболочек является то, что они позволяют непрограммистам воспользоваться результатами работы программистов, решавших аналогичные проблемы. Недостатком оболочек является их ориентированность на достаточно узкий класс задач, хотя он более широкий, чем программа, на основе которой была создана та или иная оболочка.

2. Традиционные языки программирования. Экспертные системы могут создаваться с помощью традиционных языков программирования, таких как C, C^{++}, Java, Basic, Fortran, Pascal и др. Эти языки дают проектировщику экспертных систем значительно большую свободу действий, чем оболочки. Гибкость, предоставляемая традиционными языками программирования, особенно важна в тех случаях, когда невозможно заранее определить оптимальные способы представления знаний и управления логическим выводом. Преимуществом использования традиционных языков программирования является возможность достижения высокой эффективности программного продукта, связанная с их близостью к машинной архитектуре. Еще одно достоинство — возможность включения разрабатываемых интеллектуальных подсистем в крупные программные комплексы общего назначения.

Недостатком использования традиционных языков программирования являются значительные трудозатраты программистов, обусловленные слабой приспособленностью этих языков для работы с символьными и логическими данными.

3. Языки искусственного интеллекта. К языкам искусственного интеллекта в первую очередь принято относить LISP и PROLOG. Они отличаются богатыми возможностями обработки символьных и логических данных. Язык LISP скрыл от программистов устройство памяти ЭВМ, а PROLOG позволил им не заботиться о потоке управления в программе. Естественно, что такое повышение языкового уровня привело к снижению гибкости и универсальности по сравнению с традиционными языками программирования.

Язык LISP был разработан в Стэнфордском университете в начале 1960-х гг., а PROLOG — в Марсельском университете в 1971 г. Наибольшую популярность язык PROLOG приобрел в 1980-х гг., когда благодаря усилиям математиков был обоснован логический базис этого языка, а также в силу того, что в японском проекте вычислительных систем V поколения язык PROLOG был принят в качестве базового для машины вывода.

Следует, однако, отметить, что в последнее время применение языков LISP и PROLOG несколько сократилось. Объясняется это, по-видимому, тем, что системы искусственного интеллекта часто служат модулями других больших приложений, поэтому стандарты разработки приводят к необходимости использования единого языка программирования всего приложения.

4. Интегрированные среды поддержки разработки интеллектуальных систем. Это своего рода надстройки над языками искусственного интеллекта, такие как ART, KEE, G2. Фактически они являются большим набором хорошо интегрированных парадигм искусственного интеллекта, что позволяет пользователю комбинировать в процессе разработки экспертной системы разные стили программирования.

3.5. ИНЖЕНЕРИЯ ЗНАНИЙ

При разработке экспертных систем одним из важнейших этапов является проектирование баз знаний, т. е. получение знаний специалистов предметных областей. Этот раздел искусственного интеллекта получил название *инженерии знаний.*

В настоящее время сложилось несколько стратегий получения знаний.

Наиболее ранним является способ получения знаний в результате живого контакта инженера по знаниям с экспертом либо другим источником знаний, например со специальной литературой. Многие авторы называют эту стратегию *извлечением знаний*.

Желание автоматизировать трудоемкий процесс получения знаний привело к появлению двух стратегий, основанных на применении ЭВМ: приобретение знаний и формирование знаний.

Приобретение знаний — это способ построения базы знаний посредством диалога эксперта с компьютерной программой. Обычно такие программы ориентированы на конкретные предметные области, разработаны с учетом их структуры и конкретных особенностей, т. е. они не являются универсальными. Например, одна из первых систем подобного рода — TEIRESISA — предназначена для пополнения базы знаний системы медицинской диагностики MYCIN.

Стратегия, называемая *формированием знаний,* включает применение алгоритмов автоматического порождения гипотез методом обучения на примерах принятия решений в предметной области.

Стратегии автоматизированного получения знаний (приобретение и формирование знаний) являются весьма перспективными и быстро развивающимися разделами инженерии знаний, однако до сих пор они не достигли промышленного уровня и представляют больше научный, нежели практический, интерес. Наибольшее же распространение в настоящее время по-прежнему имеет стратегия «ручного» проектирования баз знаний, т. е. извлечение знаний путем живого общения инженера по знаниям с экспертом или другим источником знаний. Это весьма трудоемкая процедура, и, как отмечают многие специалисты, она является «узким горлышком» в общем процессе создания экспертной системы.

Существует множество рекомендаций по организации работы эксперта, выбору его поведения. Знакомство с предметной областью рекомендуется начинать с изучения соответствующей литературы, учебников, монографий, статей, отчетов, сведений из сети Интернет. Затем целесообразно применение *коммуникативных* методов, подразумевающих вступление в контакт с экспертами. Коммуникативные методы разделяют на пассивные и активные. *Пассивные* включают наблюдения за объектом (технологическим процессом и др.), составляющим

предметную область, наблюдения за работой эксперта, за процессом принятия решений. К пассивным методам относится также прослушивание лекций, читаемых экспертом, изучение его инструкций и протоколов «мыслей вслух» — попыток объяснить принимаемые им решения.

Активные методы отличаются широким разнообразием ассортимента. Это анкетирование, интервью, свободный диалог, экспертные игры, дискуссии за круглым столом с участием нескольких экспертов, мозговой штурм. В результате инженер по знаниям формирует *концептуальную структуру* предметной области — модель предметной области, включающую описание ее объектов и связей между ними. Концептуальную структуру изображают в виде графов, фреймов либо описывают словами. Затем формируется *функциональная структура* предметной области — модель рассуждений эксперта и процесса принятия решений. Она представляется в виде таблицы, графа или в виде предложений естественного языка. Здесь могут присутствовать математические формулы, отражающие внутренние закономерности предметной области, а также продукционные правила, имеющие эвристическую природу, а потому снабженные коэффициентами уверенности.

Процесс проектирования базы знаний заканчивается ее формализацией и программной реализацией.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Приведите примеры предметных областей, в которых знания 1) легко поддаются формализации, 2) плохо поддаются формализации, 3) не могут быть формализованы.
2. Дайте определение и сформулируйте назначение экспертной системы.
3. Приведите примеры известных вам экспертных систем.
4. Что такое оболочка экспертной системы?
5. Какой, по вашему мнению, должен быть коллектив разработчиков экспертной системы?
6. Перечислите и охарактеризуйте стадии и этапы разработки экспертных систем.
7. Назовите отличительные признаки экспертной системы.
8. Назовите и охарактеризуйте основные стратегии получения знаний.
9. Почему задачу извлечения знаний называют «узким горлышком» в проектировании экспертных систем? Какие идеи вы можете предложить для устранения этой ситуации?

Глава 4

ПОНЯТИЕ О КЛАССИЧЕСКОЙ НЕЙРОННОЙ СЕТИ

4.1. МОЗГ И КОМПЬЮТЕР

Нейронные сети и нейрокомпьютеры — это одно из направлений компьютерной индустрии, в основе которого лежит идея создания искусственных интеллектуальных устройств «по образу и подобию» человеческого мозга. Дело в том, что большинство современных компьютеров имеет архитектуру фон Неймана. Согласно этой архитектуре, они имеют центральный процессор, запоминающее устройство, устройства ввода, устройства вывода, устройства хранения информации. По своей структуре и свойствам эти компьютеры весьма далеки от нашего естественного компьютера — человеческого мозга. В подтверждение этому в табл. 4.1 приведены признаки, отличающие человеческий мозг от неймановского компьютера.

Таблица 4.1

Сопоставление принципов построения и свойств современного компьютера (машины фон Неймана) и человеческого мозга

Признаки	Компьютер	Человеческий мозг
Процессор	Сложный	Простой
	Высокоскоростной	Низкоскоростной
	Один или несколько	Большое количество
Память	Отделена от процессора	Интегрирована в процессор
	Локализована	Распределенная
	Адресация не по содержанию	Адресация по содержанию
Вычисления	Централизованные	Распределенные
	Последовательные	Параллельные
	Хранимые программы	Самообучение
Надежность	Высокая уязвимость	Живучесть
Среда функционирования	Строго определенная	Плохо определенная
	Строго ограниченная	Без ограничений

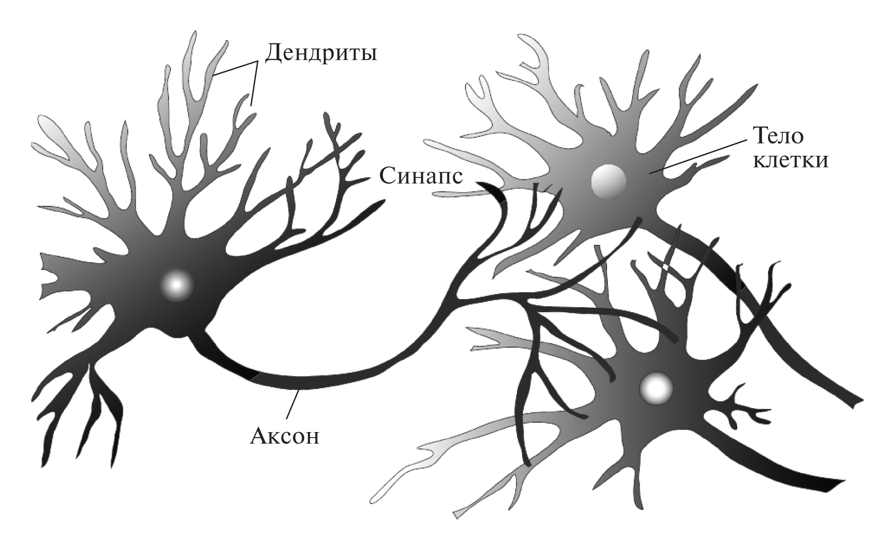


Рис. 4.1. Нейроны человеческого мозга

Основатели же нейрокибернетики[1] задались целью создать электронные устройства, адекватные мозгу не только на функциональном, но и на структурном уровне. Для этого им пришлось обратиться за сведениями к биологам. Как же устроен человеческий мозг?

Известно, что мозг человека состоит из белого и серого вещества: белое вещество — это тела нервных клеток, называемых *нейронами*, а серое вещество — соединяющие их нервные волокна. Каждый нейрон состоит из трех частей: тела клетки, дендритов и аксона (рис. 4.1). *Дендриты* и *аксон* — это нервные отростки, через которые нейрон обменивается электрическими сигналами с другими нейронами. Каждый нейрон может иметь до 10 000 дендритов и всего лишь один аксон. Через дендриты нейрон принимает электрические сигналы, поступающие от других нейронов по *нервным волокнам*, как по проводам. Если сигналов много и они достаточно интенсивны, то нейрон переходит в возбужденное состояние и сам вырабатывает электрический сигнал, который передает в аксон. Аксон на своем другом конце разветвляется на тысячи нервных волокон, которые затем соединяются с дендритами других нейронов. Места соединения нервных волокон с дендритами называются *синапсами*.

Как же человеческий мозг запоминает информацию и как ее обрабатывает? Ответить на этот вопрос биологи не могли. Но зато они знали, что общее число нейронов в течение жизни человека практически не изменяется. Это значит, что мозг ребенка и мозг взрослого человека содержат приблизительно одинаковое количество

[1] Эту стратегию создания интеллектуальных систем в гл. 1 книги мы также называли низкоуровневой, или восходящей.

нейронов. Примерно одинаковое количество нейронов содержит мозг ученого, политического деятеля, солдата, спортсмена. Отличие состоит в величинах электропроводностей синапсов.

Как известно из электротехники, электропроводность проводника ρ — это величина, обратная его электрическому сопротивлению R, т. е. $\rho = \frac{1}{R}$. Сопротивление R входит в закон Ома, $U = IR$, как коэффициент пропорциональности между приложенной к концам проводника разности потенциалов U и силой тока I, возникающего в проводнике под действием этой разности потенциалов. Чем выше электропроводность проводника ρ, тем лучше его способность проводить электрический ток.

Биологи же электропроводности синапсов называют *силами межнейронных синаптических связей*. По их мнению, мозг одного человека отличается от мозга другого человека прежде всего величинами сил межнейронных синаптических связей. На этом основании была высказана гипотеза о том, что все наши мысли, эмоции, знания, вся информация, хранящаяся в человеческом мозге, закодирована в виде огромного количества чисел, характеризующих силы межнейронных синаптических связей.

А теперь попробуем оценить, сколько же чисел способен запомнить такой гипотетический мозг, если принять, что с помощью одной синаптической связи можно закодировать одно число.

В человеческом мозге содержится приблизительно 10^{11} нейронов. Каждый нейрон связан с $10^3 \ldots 10^4$ другими нейронами. Таким образом, биологическая нейронная сеть, составляющая мозг человека, содержит $10^{14} \ldots 10^{15}$ синапсов. Получается, что именно такое количество чисел способен хранить человеческий мозг и что именно таким количеством чисел закодированы в нашем мозге все наши знания, весь жизненный опыт, все мысли и эмоции, вся информация, которую мы получаем на протяжении жизни.

Приведенные выше представления о строении и функционировании мозга в настоящее время считаются научно обоснованным фактом. Ни у кого из ученых не вызывает сомнений, что разум человека создается огромным количеством мельчайших нервных клеток — нейронами, непрерывно исполняющими свой «*информационный танец*». Что это за «танец», мы рассмотрим в следующих параграфах.

4.2. МАТЕМАТИЧЕСКИЙ НЕЙРОН МАК-КАЛЛОКА—ПИТТСА

Первой работой, которая заложила теоретический фундамент для создания интеллектуальных устройств, моделирующих человеческий мозг на самом низшем — структурном — уровне, принято считать

опубликованную в 1943 г. статью *Уоррена Мак-Каллока* и *Уолтера Питтса* «Идеи логических вычислений в нервной деятельности» [100]. Ее авторы, американские ученые математики-нейрофизиологи, по праву считаются основателями нейроинформатики. Они предложили математическую модель нейрона мозга человека, назвав ее *математическим* или *модельным нейроном*.

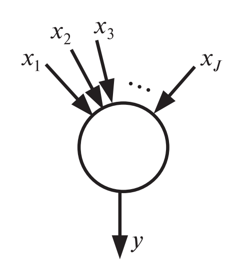


Рис. 4.2. Математический нейрон Мак-Каллока—Питтса

У. Мак-Каллок и В. Питтс предложили изображать нейрон в виде кружочка со стрелочками, как показано на рис. 4.2. Стрелки означают входы и выход нейрона. Через входы математический нейрон принимает *входные сигналы* $x_1, x_2, \ldots, x_j, \ldots, x_J$ и суммирует их, умножая каждый входной сигнал на некоторый *весовой коэффициент* w_j:

$$S = \sum_{j=1}^{J} w_j x_j. \tag{4.1}$$

После выполнения операции суммирования математический нейрон формирует выходной сигнал y согласно следующему правилу:

$$y = \begin{cases} 1, & \text{если } S \geqslant \theta; \\ 0, & \text{если } S < \theta, \end{cases} \tag{4.2}$$

где θ — *порог чувствительности нейрона*.

Таким образом, математический нейрон, как и его биологический прототип — нейрон мозга, может существовать в двух состояниях — возбужденном и невозбужденном. Если взвешенная сумма входных сигналов S меньше пороговой величины θ, то математический нейрон не возбужден и его выходной сигнал равен нулю. Если же входные сигналы достаточно интенсивны и их сумма достигает порога чувствительности θ, то нейрон переходит в возбужденное состояние и на его выходе формируется сигнал $y = 1$.

Весовые коэффициенты w_j имеют определенный физический смысл. Они имитируют электропроводности нервных волокон, тех самых, которые биологи называют *силами межнейронных синаптических связей*, или *синаптическими весами*. Чем эти силы больше, тем большей величины сигналы попадают в нейрон и тем выше вероятность его перехода в возбужденное состояние.

Ранее мы уже отмечали, что биологи придают силам синаптических связей очень важное значение. Многие из них полагают, что именно с их помощью человеческий мозг кодирует всю имеющуюся у него информацию. Как мы увидим далее, в искусственном мозге — в нейронных сетях и в нейрокомпьютерах — происходит то же самое. Вся информация в них тоже кодируется в виде множества чисел, характеризующих силы межнейронных синаптических связей w_j.

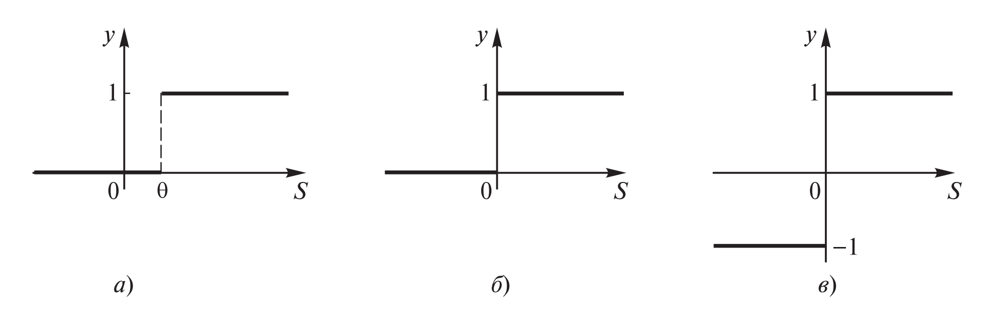


Рис. 4.3. Пороговые активационные функции нейрона, заданные формулами: *а* — (4.2); *б* — (4.4); *в* — (4.5)

Логическая функция (4.2) называется *активационной функцией нейрона*. Ее графическое изображение, представленное на рис. 4.3, *а*, по форме напоминает ступеньку, поэтому ее называют *функцией-ступенькой*.

Таким образом, математический нейрон представляет собой пороговый элемент с несколькими входами и одним выходом. Каждый математический нейрон имеет свое определенное значение порога чувствительности θ.

Авторы математического нейрона У. Мак-Каллок и В. Питтс в своей статье [100] также показали, что с помощью математического нейрона можно моделировать различные логические функции, например функцию логического умножения «И» (ее также обозначают «AND»), функцию логического сложения «ИЛИ» («OR») и функцию логического отрицания «НЕТ» («NOT»). Таблицы истинности этих логических функций приведены в табл. 4.2, в которых значение логических функций «истинно» закодировано единицей, а значение «ложно» — нулем.

С помощью этих таблиц и формул (4.1)–(4.2) нетрудно убедиться, что математический нейрон (рис. 4.4), имеющий два входа с единичными силами синаптических связей $w_1 = w_2 = 1$, моделирует функцию логического умножения «И» при $\theta = 2$, Этот же нейрон моделирует функцию логического сложения «ИЛИ» при задании $\theta = 1$. Матема-

Таблица 4.2

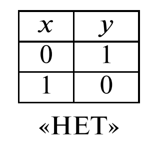
Таблицы истинности логических функций

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

«И»

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

«ИЛИ»



x	y
0	1
1	0

«НЕТ»

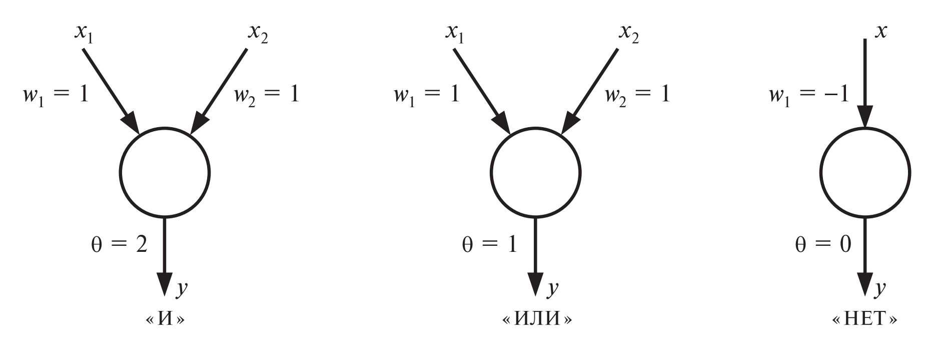


Рис. 4.4. Математические нейроны, моделирующие логические функции

тический нейрон с одним входом моделирует функцию «НЕТ» при задании $w = -1$ и $\theta = 0$.

В современной литературе иногда вместо понятия порога чувствительности нейрона θ используют термин *нейронное смещение* b, которое отличается от θ только знаком: $b = -\theta$. Если его добавить к сумме (4.1):

$$S = \sum_{j=1}^{J} w_j x_j + b, \tag{4.3}$$

то пороговая активационная функция нейрона примет вид:

$$y = \begin{cases} 1, & \text{если } S \geqslant 0; \\ 0, & \text{если } S < 0. \end{cases} \tag{4.4}$$

Графическое представление этой активационной функции приведено на рис. 4.3, *б*. Еще более симметричный вид, представленный на рис. 4.3, *в*, активационная функция нейрона приобретает при использовании формулы:

$$y = \begin{cases} 1, & \text{если } S \geqslant 0; \\ -1, & \text{если } S < 0. \end{cases} \tag{4.5}$$

В формуле (4.3) нейронное смещение b можно рассматривать как вес w_0 некоторого дополнительного входного сигнала x_0, величина которого всегда равна единице:

$$S = \sum_{j=1}^{J} w_j x_j + w_0 x_0 = \sum_{j=0}^{J} w_j x_j. \tag{4.6}$$

Нейрон с дополнительным входом x_0 изображен на рис. 4.5.

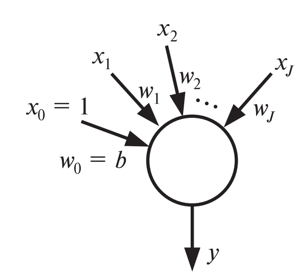


Рис. 4.5. Нейронное смещение b интерпретируется как вес дополнительного входа с силой синаптической связи w_0, сигнал которого x_0 всегда равен единице

Подводя итог изложенному, еще раз отметим, что, согласно наиболее распространенным современным научным представлениям, вся

информация и все знания в человеческом мозге кодируются и хранятся в виде матрицы сил межнейронных синаптических связей. На языке электротехники сила межнейронной синаптической связи — это электропроводность синапса, который представляет собой точку электрохимического контакта между дендритом нейрона и нервным волокном. Математический нейрон Мак-Каллока — Питтса — это математическая модель биологического нейрона мозга, учитывающая его структуру и функциональные свойства.

Контрольные вопросы и задания к § 4.2

1. Назовите несколько отличительных признаков в принципах действия современного компьютера, выполненного по схеме фон Неймана, от мозга.
2. Сколько нейронов имеет человеческий мозг?
3. Сколько дендритов и сколько аксонов может иметь нейрон? Каково их назначение?
4. В каком виде хранится информация в человеческом мозге?
5. Объясните на языке электротехники значение термина «сила синаптической связи». В каких единицах она измеряется?
6. Какой объем памяти имеет человеческий мозг? Сколько чисел он может запомнить?
7. Напишите формулы, с помощью которых происходит преобразование сигналов в математическом нейроне Мак-Каллока—Питтса.
8. Нарисуйте графическое изображение активационной функции математического нейрона Мак-Каллока—Питтса.
9. Нарисуйте математические нейроны, реализующие логические функции «И», «ИЛИ», «НЕТ» и приведите соответствующие им значения сил синаптических связей и порогов.
10. Зайдите на сайт **www.LbAi.ru** и выполните лабораторную работу № 1 — обучите математический нейрон моделировать функции «И» и «ИЛИ». В случае затруднений или заинтересованности обратитесь к учебно-методическим пособиям [73, 88].

4.3. ПЕРСЕПТРОН РОЗЕНБЛАТТА И ЕГО ОБУЧЕНИЕ

Как отмечалось ранее, американские ученые У. Мак-Каллок и В. Питтс предложили математическую модель нейрона мозга человека, назвав ее математическим нейроном. Так же как и биологический нейрон, математический нейрон имеет несколько входов и один выход, может существовать в возбужденном и невозбужденном состояниях, причем переход в возбужденное состояние зависит от величины поступающих к нему сигналов и сил межнейронных синаптических связей. Таким образом, математический нейрон весьма правдоподобно имити-

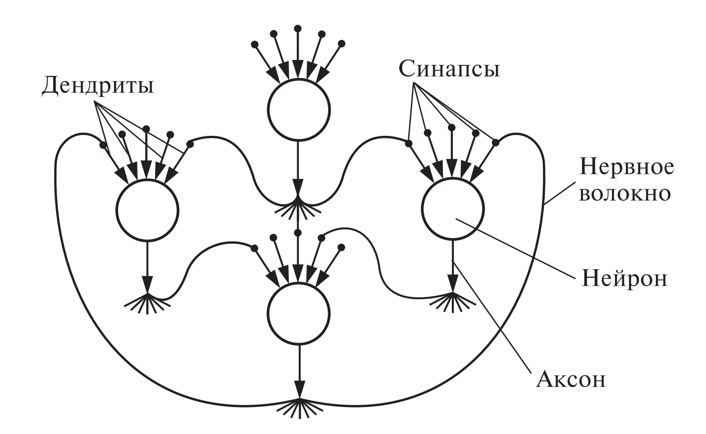


Рис. 4.6. Математические нейроны, связанные между собой проводами в нейронную сеть

рует структуру и свойства своего прототипа — биологического нейрона мозга. На этом основании У. Мак-Каллок и В. Питтс в своей статье [100] высказали весьма смелую и даже несколько фантастическую гипотезу, которая впоследствии легла в основу современной нейроинформатики. Они предположили, что если математические нейроны связать между собой проводами, имитирующими нервные волокна мозга (рис. 4.6), и пустить по проводам электрические сигналы, как это происходит в мозге, то такой искусственный мозг будет способен решать интеллектуальные задачи подобно тому, как это делает естественный человеческий мозг!

И эта идея, которую критики называли абсурдной, через 15 лет была блестяще подтверждена американским ученым ***Фрэнком Розенблаттом*** [48, 101, 102]. В 1958 г. он создал компьютерную программу для IBM-794, эмулирующую деятельность математических нейронов. Это была первая ***нейронная сеть*** или сокращенно — ***нейросеть***. Она была названа ***персептроном*** от английского слова *perception* — осознание.

Затем, спустя два года, Розенблатт смонтировал электронное устройство, в котором функции математических нейронов выполняли отдельные электросхемы, работающие на электронных лампах. Это был первый *нейрокомпьютер*, который успешно решал сложнейшую интеллектуальную задачу — распознавал буквы латинского алфавита, изображенные на карточках, подносимых к его считывающему устройству — электронному глазу.

Итак, смелую гипотезу Мак-Каллока — Питтса удалось подтвердить экспериментальным путем. Но раз эксперимент удался, значит, правильными оказались наши представления о биологической структуре и строении мозга, о его внутренних электрофизиологических процессах, о способе запоминания и хранения информации. Бы-

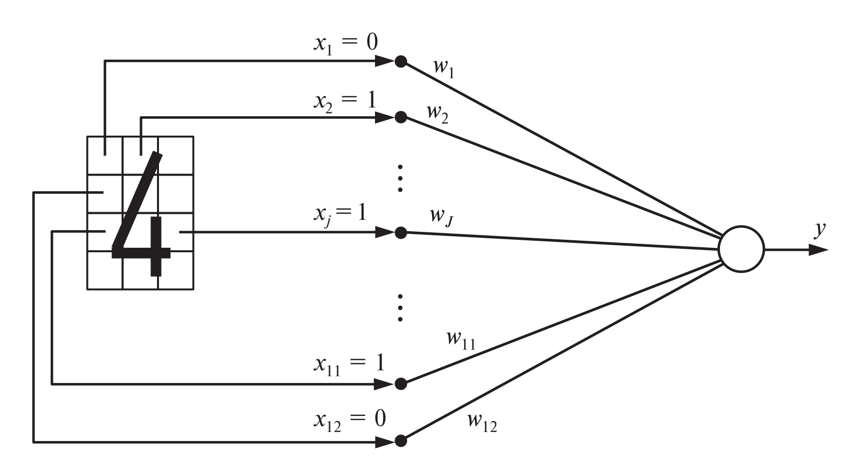


Рис. 4.7. Персептрон, классифицирующий числа на четные и нечетные

ла подтверждена адекватность математического нейрона, как модели биологического нейрона. Была подтверждена адекватность нейросети и нейрокомпьютера, как модели мозга. «***Нельзя сказать, что мы точно воспроизводим работу человеческого мозга***, — писал Ф. Розенблатт, — ***но пока персептрон ближе всего к истине***».

Разберем принцип действия персептрона на примере решения конкретных задач. На рисунке 4.7 приведен один из простейших вариантов исполнения персептрона, предназначенного для классификации чисел на четные и нечетные. Представим себе матрицу из 12 фотоэлементов, расположенных в виде четырех горизонтальных рядов по три фотоэлемента в каждом ряду. На матрицу фотоэлементов накладывается карточка с изображением цифры, например, «4» (см. рис. 4.7). Если на какой-либо фотоэлемент попадает фрагмент цифры, то этот фотоэлемент вырабатывает сигнал в виде единицы, в противном случае — нуль. На рис. 4.7 на первый фотоэлемент не попал фрагмент цифры, и поэтому его сигнал $x_1 = 0$; на второй фотоэлемент попал фрагмент цифры, и поэтому он вырабатывает сигнал $x_2 = 1$ и т. д.

Согласно формулам (4.1)–(4.2), математический нейрон выполняет суммирование входных сигналов x_j, помноженных на синаптические веса w_j. Затем результат суммирования S сравнивается с порогом чувствительности θ и вырабатывается выходной сигнал y.

Первоначальные значения синаптических весов w_j и порога чувствительности θ Розенблатт задавал датчиком случайных чисел, поэтому на выходе персептрона случайным образом вырабатывался сигнал: либо 0, либо 1.

Задача состояла в следующем. Требовалось подобрать значения синаптических весов w_j такими, чтобы выходной сигнал y принимал

значение единица, если на карточке было изображено четное число, и нуль, если число было нечетным.

Эту задачу Ф. Розенблатт решил путем поочередного накладывания на фотоэлементы карточек и *обучения* персептрона путем корректировки синаптических весов w_j. Если, например, на вход персептрона предъявлялась карточка с цифрой «4» и выходной сигнал y случайно оказывался равным единице, означающей четность, то корректировать синаптические веса было не нужно, так как реакция персептрона правильна. А если выходной сигнал оказался равным нулю, что неправильно, то следовало увеличить (поощрить) веса тех активных входов, которые способствовали возбуждению нейрона. В данном случае увеличению подлежали w_2, w_{11} и др.

Следуя этой идее, можно сформулировать *итерационный алгоритм* корректировки синаптических весов, обеспечивающий обучение персептрона в нужном направлении.

Шаг 1. Датчиком случайных чисел всем синаптическим весам w_j ($j = 1, \ldots, 12$) и порогу чувствительности нейрона θ присвоить некоторые малые случайные значения.

Шаг 2. Предъявить персептрону какую-либо цифру. Системой фотоэлементов вырабатывается входной вектор x_j ($j = 1, \ldots, 12$).

Шаг 3. Нейрон выполняет взвешенное суммирование входных сигналов

$$S = \sum_{j=1}^{12} w_j x_j$$

и вырабатывает выходной сигнал $y = 1$, если $S \geqslant \theta$, или $y = 0$, если $S < \theta$.

Шаг 4, а. Если выходной сигнал правильный, то перейти на *шаг 2*.

Шаг 4, б. Если выходной сигнал неправильный и равен нулю, то увеличить веса активных входов: например, добавить каждому j-му синаптическому весу величину j-го входного сигнала

$$w_j(t+1) = w_j(t) + x_j.$$

Тогда, если вход был неактивен, т. е. $x_j = 0$, то j-й синаптический вес не изменится. Если же вход был активен, т. е. $x_j = 1$, то j-й синаптический вес будет увеличен на единицу.

Здесь и далее t означает номер итерации, которые в искусственном интеллекте называют *эпохами*; $w_j(t+1)$ — новое значение (на новой эпохе) j-го синаптического веса; $w_j(t)$ — его старое значение (на предыдущей эпохе).

Шаг 4, в. Если выходной сигнал неправильный и равен единице, то уменьшить веса активных входов, например, с помощью аналогичной формулы:

$$w_j(t+1) = w_j(t) - x_j.$$

Шаг 5. Перейти на *шаг 2* или завершить процесс обучения.

В приведенном здесь алгоритме *шаг 4, б* называют ***первым правилом Хебба***, а *шаг 4, в* — ***вторым правилом Хебба*** в честь канадского ученого физиолога *Д. О. Хебба*, предложившего этот алгоритм в 1949 г. [94].

Отметим, что алгоритм обучения персептрона с помощью правил Хебба удивительным образом напоминают процесс обучения ребенка или студента методом «поощрения — наказания» (или дрессировки животного методом «кнута и пряника»). Обратим внимание также на то, что первоначальные значения синаптических весов w_j задаются датчиком случайных чисел. Это соответствует тому, что при рождении человека или животного его мозг еще не накопил знаний, и поэтому силы синаптических связей w_j имеют какие-то случайные значения. Как и в случаях с ребенком, студентом и животным, обучаемом методом «поощрения-наказания», алгоритм обучения персептрона за конечное число попыток (их называют ***итерациями***, или ***эпохами***) может привести к цели — персептрон в конце концов усвоит необходимые знания, закодирует их в виде конкретных значений матрицы сил синаптических связей w_j и, таким образом, научится различать четные и нечетные числа.

Рассмотренный выше алгоритм обучения персептрона можно представить в более общей форме. Если через d обозначить требуемый выходной сигнал (от слов *desire response*, что в переводе с английского означает — желаемый отклик), то на каждой эпохе обучения можно определять разницу между требуемым ответом персептрона d и реальным значением y, вычисляемым на его выходе:

$$\varepsilon = d - y.$$

Тогда:

- случай $\varepsilon = 0$ соответствует шагу 4, *а*;
- случай $\varepsilon > 0$ соответствует шагу 4, *б*;
- случай $\varepsilon < 0$ соответствует шагу 4, *в*.

Идея алгоритма обучения персептрона с помощью правил Хебба сохранится, если итерационный процесс корректировки синаптических весов вести по формулам:

$$w_j(t+1) = w_j(t) + \Delta w_j; \quad (4.7)$$

$$\Delta w_j = \varepsilon x_j, \quad (4.8)$$

где $w_j(t)$ и $w_j(t+1)$ — старое и новое значения весовых коэффициентов персептрона; j — номер входного сигнала.

Кроме того, можно получить аналогичную итерационную формулу для подстройки нейронного смещения b, если учесть, что его можно интерпретировать как вес w_0 дополнительного входа x_0, значение которого равно единице (см. рис. 4.5 и формулы (4.3)–(4.6)):

$$w_0(t+1) = w_0(t) + \Delta w_0; \quad (4.9)$$

$$\Delta w_0 = \varepsilon. \quad (4.10)$$

В итерационные формулы полезно ввести *коэффициент скорости обучения* η, с помощью которого можно управлять величиной коррекции синаптических весов и нейронного смещения:

$$\Delta w_j = \eta \varepsilon x_j; \tag{4.11}$$

$$\Delta w_0 = \eta \varepsilon. \tag{4.12}$$

При $\eta = 1$ коэффициент скорости обучения не влияет на итерационный процесс. При $\eta > 1$ процесс обучения ускоряется, однако при слишком больших значениях коэффициента η итерационный процесс может потерять устойчивость и разойтись. При $\eta < 1$ итерационный процесс обычно стабилизируется, но при этом могут чрезмерно возрасти временны́е затраты. На практике коэффициент скорости обучения η задают в пределах от 0,05 до 1,5.

Алгоритм обучения персептрона с использованием этих формул известен под названием *дельта-правила*.

Естественно, возникает вопрос, всегда ли алгоритм обучения персептрона приводит к желаемому результату. Ответ на этот вопрос дает *теорема сходимости персептрона*:

Если существует множество значений весов, которые обеспечивают требуемое распознавание образов, то в конечном итоге алгоритм обучения персептрона приводит либо к этому множеству, либо к другому множеству, такому, что требуемое распознавание образов будет достигнуто.

Как следует из этой теоремы, задача нахождения матрицы весовых коэффициентов w_j, обеспечивающих распознавание образов, может иметь множество решений — таких матриц может быть много. В то же время, в формулировке теоремы не говорится, что такие матрицы всегда существуют, и значит, не всегда существует решение задачи.

В настоящее время считается, что по числу выполненных доказательств теорема сходимости персептрона занимает первое место в мире. Ранее самой доказанной в мире теоремой считалась теорема Пифагора.

Контрольные вопросы и задания к § 4.3

1. Чем отличается нейронная сеть от нейрокомпьютера?
2. Каким образом вырабатываются входные сигналы x_1, x_2, x_3… персептрона, классифицирующего числа на четные и нечетные?
3. Каким образом задаются первоначальные значения синаптических весов w_1, w_2, w_3,… и как они затем корректируются?
4. В каком виде персептрон хранит знания, необходимые для распознавания цифр?
5. Что подразумевается под введенными нами обозначениями d и y? Чем они отличаются?

6. Напишите формулы, по которым, согласно алгоритму дельта-правила, корректируются синаптические веса и нейронные смещения.
7. Какая теорема считается самой доказанной в мире теоремой?
8. Дайте формулировку теоремы сходимости персептрона.
9. Зайдите на сайт **www.LbAi.ru** и выполните лабораторную работу № 2 — обучите персептрон классифицировать числа на четные и нечетные. В случае затруднений или заинтересованности обратитесь к учебно-методическим пособиям [73, 88].

4.4. РАСПОЗНАВАНИЕ БУКВ

Дальнейшее развитие идеи персептрона и алгоритмов обучения связано с усложнением его структуры и функциональных свойств. На рисунке 4.8 представлена схема персептрона, предназначенного для распознавания букв русского алфавита. В отличие от предыдущей схемы такой персептрон имеет 33 выходных нейрона: каждой букве алфавита соответствует свой выходной нейрон. Полагается, что сигнал первого выходного нейрона y_1 должен быть равен единице, если персептрону предъявлена буква «А», и равен нулю, если предъявляется любая другая буква. Выход второго нейрона y_2 должен быть равен единице, если персептрону предъявлена буква «Б», и равен нулю во всех остальных случаях. И так далее до буквы «Я».

Алгоритм обучения данного персептрона выглядит следующим образом.

Шаг 1. Датчиком случайных чисел всем весовым коэффициентам w_{ij} и нейронным смещениям w_{i0} ($i = 1, \ldots, 33$, $j = 1, \ldots, 12$) присваиваются некоторые малые случайные значения.

Шаг 2. Персептрону предъявляется какая-либо буква алфавита, системой фотоэлементов вырабатывается входной вектор x_j ($j = 1, \ldots, 12$). Сигналы дополнительных нейронных входов присваиваются единичными: $x_0 = 1$.

Шаг 3. Каждый нейрон выполняет взвешенное суммирование входных сигналов

$$S_i = \sum_{j=0}^{12} w_{ij} x_j$$

и вырабатывает выходной сигнал $y_i = 1$, если $S_i \geqslant 0$; $y_i = 0$, если $S_i < 0$.

Шаг 4. Для каждого нейрона вычисляется его *ошибка*

$$\varepsilon_i = d_i - y_i,$$

где d_i — вектор правильных (желаемых) ответов персептрона, например для буквы «А» $d_1 = 1$, $d_2 = 0$, …, $d_{33} = 0$ и т. д.

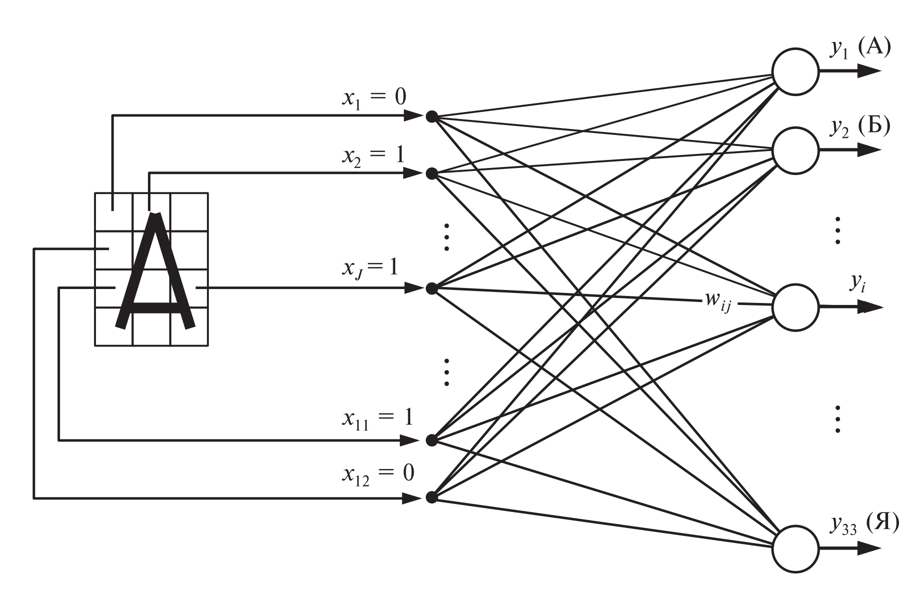


Рис. 4.8. Персептрон, предназначенный для распознавания букв русского алфавита

Шаг 5. Производится корректировка весовых коэффициентов и нейронных смещений:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}; \quad \Delta w_{ij} = \eta \varepsilon_i x_j;$$
$$w_{i0}(t+1) = w_{i0}(t) + \Delta w_{i0}; \quad \Delta w_{i0} = \eta \varepsilon_i,$$

где t — номер эпохи; $i = 1, \ldots, 33$; $j = 1, \ldots, 12$.

Шаг 6. Повторение *шагов 2—5* необходимое количество раз.

Заметим, что в этом алгоритме формулы для корректировки нейронных смещений w_{i0} можно не писать, так как они будут выполняться автоматически, если цикл по индексу j начинать не от единицы, а от нуля.

Как уже отмечалось ранее, первый действующий персептрон был создан в 1958–1961 гг. Он был предназначен для распознавания букв латинского алфавита. Буквы, отпечатанные на карточках, поочередно накладывали на табло фотоэлементов и осуществляли процесс обучения персептрона согласно приведенному здесь алгоритму. После выполнения достаточно большого количества эпох персептрон научился безошибочно распознавать все буквы, участвовавшие в обучении. Таким образом, была подтверждена гипотеза о том, что компьютер, построенный по образу и подобию человеческого мозга, может решать интеллектуальные задач и, в частности, решать задачу распознавания образов — букв латинского алфавита.

Но это было не все. Помимо того, что персептрон научился распознавать знакомые образы, т. е. те образы, которые демонстрировались ему в процессе обучения, он успешно справлялся с распознавани-

ем образов, которые «видел» впервые. Выяснилось, что персептрон оказался способным распознавать буквы, отпечатанные с небольшими искажениями и даже другим шрифтом, если шрифт не слишком сильно отличался от используемого при обучении персептрона.

Свойство мозга узнавать образы, которые ему встретились впервые, называется свойством *обобщения*. Это свойство было унаследовано персептроном непосредственно от его прототипа — мозга. Оно было унаследовано благодаря тому, что персептрон является адекватной моделью мозга, удачно отражающей как его структурные, так и функциональные качества. Именно свойство *обобщения* впоследствии позволило применять нейронные сети для решения широчайшего круга практических задач, недоступных для традиционных методов информатики. Именно благодаря этому свойству нейронные сети стали эффективнейшим инструментом научных исследований и практических приложений. Именно благодаря этому свойству нейросетевые и нейрокомпьютерные технологии заняли то лидирующее положение, которое они занимают в настоящее время.

Контрольные вопросы и задания к § 4.4

1. Что такое коэффициент скорости обучения, для чего он нужен и в каких пределах его обычно задают? К чему приводят слишком большие и слишком маленькие значения скорости обучения?
2. Чем отличается схема персептрона, предназначенного для классификации чисел на четные и нечетные, от схемы персептрона, распознающего буквы русского алфавита?
3. Какое количество выходных нейронов должен иметь персептрон, предназначенный для распознавания не только букв, но и цифр?
4. Что понимается под свойством обобщения, которым обладает мозг человека и его модель — персептрон?
5. Как научить персептрон распознавать не только печатные, но и рукописные буквы?
6. Зайдите на сайт **www.LbAi.ru** и выполните лабораторные работы № 3, 4 — обучите персептрон распознавать буквы алфавита. В случае затруднений или заинтересованности обратитесь к учебно-методическим пособиям [73, 88].

4.5. СИГМОИДНАЯ АКТИВАЦИОННАЯ ФУНКЦИЯ И ОБОБЩЕННОЕ ДЕЛЬТА-ПРАВИЛО

Персептрон, схема которого приведена на рис. 4.8, предназначен для распознавания букв алфавита. Можно попытаться использовать его для решения других практических задач, например ставить диагнозы болезней или определять: свой или чужой самолет подлетает к гра-

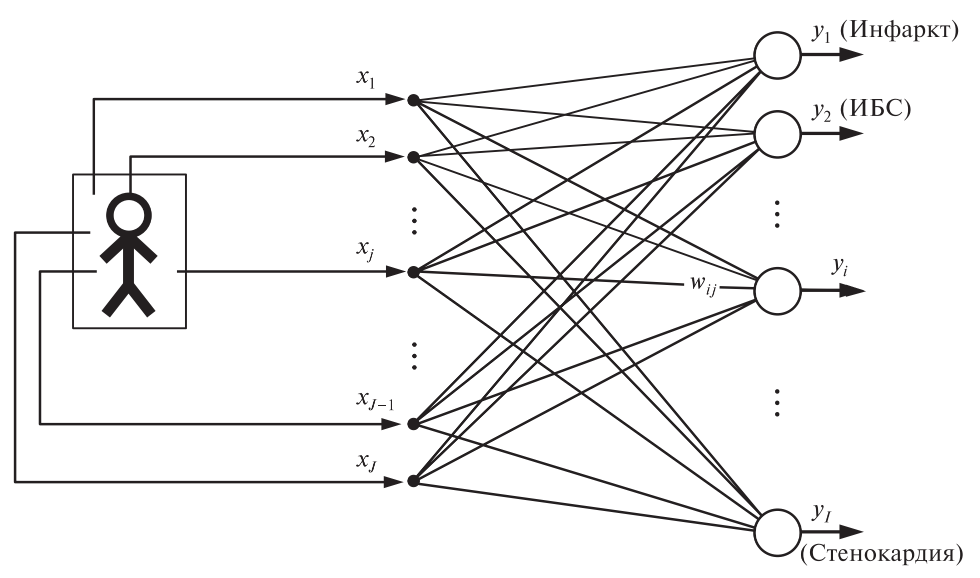


Рис. 4.9. Персептрон, предназначенный для медицинской диагностики

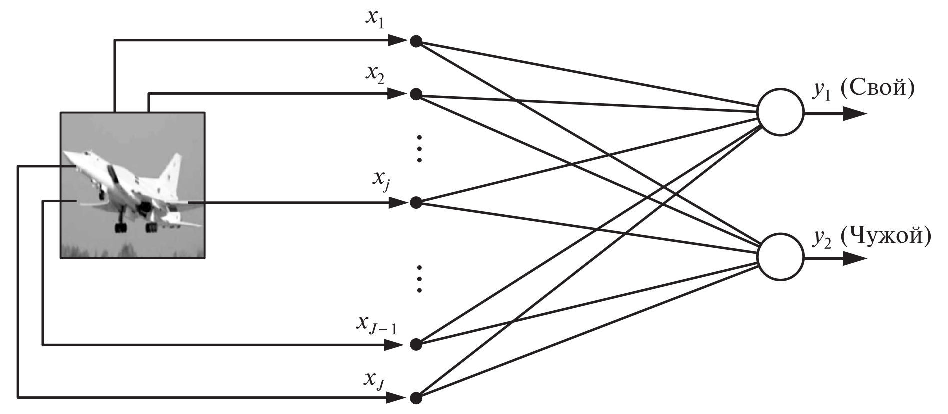


Рис. 4.10. Персептрон, предназначенный для распознавания военных объектов

ницам страны. Все зависит от того, какой смысл придавать входному вектору x_j и выходному вектору y_i.

Так, например, если в качестве x_j на вход персептрона подавать сигналы, кодирующие симптомы заболевания человека, а в качестве y_i на выходе персептрона снимать сигналы, кодирующие диагнозы его заболеваний (рис. 4.9), то на основе такого персептрона можно построить систему медицинской диагностики.

А если в качестве x_j на вход персептрона подавать сигналы с радара, который обнаружил подлетающий к границам страны самолет (рис. 4.10), а в качестве y_i на выходе персептрона снимать сигналы,

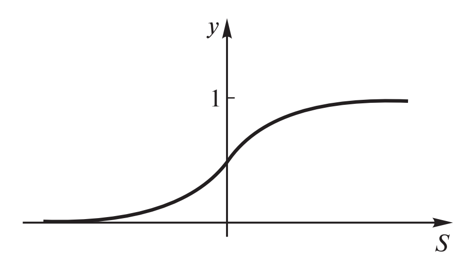


Рис. 4.11. Сигмоидная активационная функция $y = f_\sigma(S)$

кодирующие информацию о том, является ли запеленгованный самолет «своим» или «чужим», то на основе такого персептрона можно создать систему обнаружения-распознавания военных объектов.

Однако следует заметить, что для других классов задач, например прогнозирование погоды, температуры воздуха, прогнозирование котировок акций и курсов валют, такой персептрон не годится, так как он может выдавать только бинарные результаты типа «нуль» и «единица».

Круг решаемых задач значительно расширится, если научить персептрон выдавать не только бинарные выходные сигналы, но и аналоговые, т. е. имеющие непрерывные значения. Такое развитие персептрона было сделано американскими учеными *Б. Уидроу* и *М. Е. Хоффом* [107], которые вместо ступенчатой активационной функции (см. рис. 4.3) ввели непрерывную

$$y = \frac{1}{1 + e^{-S}}, \tag{4.13}$$

график которой изображен на рис. 4.11.

Эту функцию назвали *сигмоидой* за то, что ее графическое изображение напоминает латинскую букву «S». Другое название сигмоиды — *логистическая функция*. Для сигмоиды приняли обозначение $y = f_\sigma(S)$.

Подобно обычной пороговой функции активации, сигмоида отображает точки области определения $(-\infty, +\infty)$ в значения из интервала $(0, +1)$. Практически сигмоида обеспечивает непрерывную аппроксимацию классической пороговой функции.

Появление персептронов с непрерывными активационными функциями обусловило появление новых подходов к их обучению. Б. Уидроу и М. Е. Хофф предложили минимизировать квадратичную ошибку, определяемую формулой:

$$\varepsilon = \frac{1}{2} \sum_{i=1}^{I} (d_i - y_i)^2, \tag{4.14}$$

в которой, как и раньше, d_i — требуемый (желаемый) выход i-го нейрона, а y_i — выход, который получился в результате вычислений персептрона.

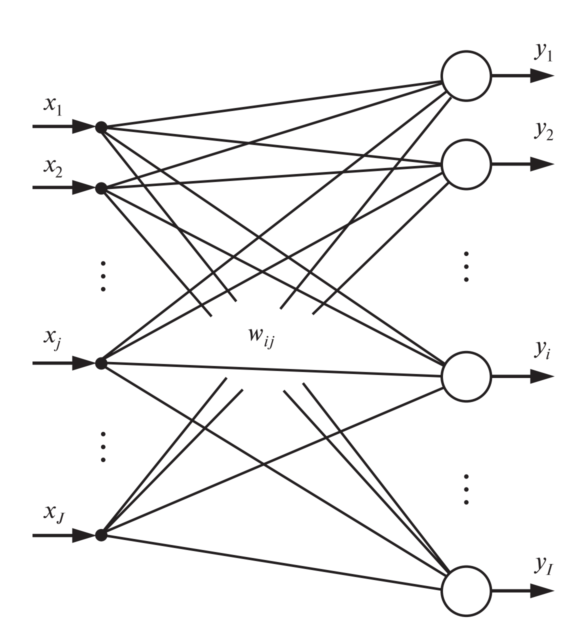


Рис. 4.12. Персептрон с J входами и I выходами

Рассмотрим алгоритм коррекции весовых коэффициентов персептрона, имеющего J входов и I выходов (рис. 4.12).

Квадратичная ошибка обучения персептрона ε зависит от того, какими являются весовые коэффициенты w_{ij}. Другими словами, ε является функцией от множества весовых коэффициентов: $\varepsilon = \varepsilon(w_{ij})$. Для ее графического представления требуется многомерная система координат, которую мы в нашем трехмерном мире представить себе не можем. В этой многомерной системе координат функция $\varepsilon = \varepsilon(w_{ij})$ изображается в виде многомерной поверхности, называемой *гиперповерхностью*.

Чтобы хоть как-то представить себе гиперповерхность, предположим, что все аргументы w_{ij} «заморожены», т. е. не меняются, за исключением двух, например $w_{i,j}$ и $w_{i,j+1}$, которые являются переменными. Тогда в трехмерной системе координат ($w_{i,j}, w_{i,j+1}$, ε) гиперповерхность будет иметь вид фигуры, напоминающей параболоид, которую назовем *псевдопарабалоидом* (рис. 4.13). Процесс обучения персептрона теперь можно представить как отыскание такого сочетания весовых коэффициентов w_{ij}, которому соответствует самая нижняя точка *гиперпсевдопараболоида*. Задачи подобного рода называются *оптимизационными*. Говорят, что оптимизационная задача состоит в *минимизации функции* $\varepsilon = \varepsilon(w_{ij})$ в многомерном пространстве параметров w_{ij}.

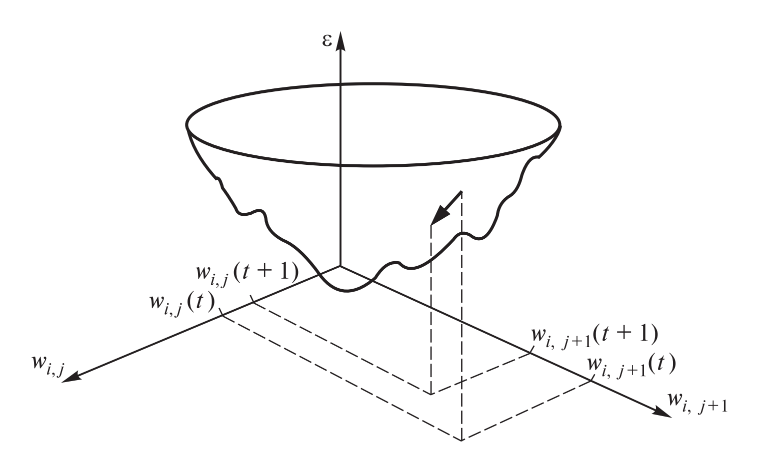


Рис. 4.13. Графическое изображение функции-ошибки персептрона $\varepsilon = \varepsilon(w_{ij})$ в трехмерной системе координат $w_{i,j}$, $w_{i,j+1}$, ε

Таким образом, если раньше мы говорили, что персептрон обучают методом «поощрения — наказания», то теперь мы будем говорить, что задача обучения персептрона — это задача оптимизации (минимизации) функции-ошибки персептрона $\varepsilon = \varepsilon(w_{ij})$. Иногда ее называют погрешностью персептрона.

Существует множество методов решения оптимизационных задач. Наиболее простым методом является перебор весовых коэффициентов w_{ij} с последующими вычислениями и сравнениями между собой значений функции ε, соответствующих этим коэффициентам. Однако более эффективны так называемые *градиентные методы*.

Градиент функции является очень важным математическим понятием, с которым обычно знакомятся на первых курсах вузов. Напомним, что градиент функции $\varepsilon = \varepsilon(w_{ij})$ представляет собой вектор, проекциями которого на оси координат являются частные производные от функции ε по этим координатам $\partial\varepsilon/\partial w_{ij}$, и что градиент функции всегда направлен в сторону ее наибольшего возрастания. Поскольку задача состоит в отыскании минимума функции $\varepsilon = \varepsilon(w_{ij})$, то нам надо опускаться по поверхности ошибок, что обеспечивается движением в сторону, противоположную градиенту этой функции. Отсюда название — *метод градиентного спуска*.

Движение в сторону, противоположную градиенту, будет осуществляться, если на каждой эпохе к координатам текущей точки w_{ij} мы, используя знакомую нам из прошлого параграфа итерационную формулу $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}$, будем добавлять величину, прямо пропорциональную частной производной по координате w_{ij}, взятую

с противоположным знаком:

$$\Delta w_{ij} = -\eta \frac{\partial \varepsilon}{\partial w_{ij}} . \tag{4.15}$$

Здесь η — некоторый коэффициент, обычно задаваемый в пределах от 0,05 до 1,5 и называемый, как и раньше, *коэффициентом скорости обучения*.

Обратите внимание, что, согласно формуле (4.15), мы движемся не только в сторону убывания функции, но и со скоростью, прямо пропорциональной скорости убывания (т. е. крутизне) функции, так как делаем шаг Δw_{ij}, пропорциональный производной, взятой со знаком минус.

Квадратичная ошибка ε является сложной функцией, зависящей от выходных сигналов персептрона y_i, которые, в свою очередь, зависят от w_{ij}, т. е. $\varepsilon = \varepsilon(y_i(w_{ij}))$. По правилу дифференцирования сложной функции

$$\frac{\partial \varepsilon}{\partial w_{ij}} = \frac{\partial \varepsilon}{\partial y_i} \frac{\partial y_i}{\partial w_{ij}} . \tag{4.16}$$

Выходные сигналы нейронов y_i вычисляются с помощью сигмоидных активационных функций $y_i = f_\sigma(S_i)$, аргументом которых являются суммы $S_i = \sum\limits_{j=1}^{J} w_{ij}x_j$, или, что то же самое: $S_i = w_{i1}x_1 + w_{i2}x_2 + \ldots + w_{ij}x_j + \ldots + w_{iJ}x_J$. Следовательно,

$$\frac{\partial y_i}{\partial w_{ij}} = \frac{\partial f_\sigma(S_i)}{\partial S_i} \frac{\partial S_i}{\partial w_{ij}} = f'_\sigma(S_i)x_j. \tag{4.17}$$

Аналогичным образом, вспоминая формулу (4.14) и выполняя дифференцирование ε по y_i, получаем:

$$\frac{\partial \varepsilon}{\partial y_i} = -(d_i - y_i). \tag{4.18}$$

Подставив (4.17) и (4.18) в (4.16) и затем полученное выражение в (4.15), окончательно будем иметь

$$\Delta w_{ij} = -\eta \left(-(d_i - y_i)f'_\sigma(S_i)x_j\right) = \eta(d_i - y_i)f'_\sigma(S_i)x_j. \tag{4.19}$$

Это выражение получено для нейронов с активационными функциями любого вида. Если $f_\sigma(S_i)$ — сигмоида, заданная формулой (4.13), то

$$f'_\sigma(S_i) = \left((1 + \mathrm{e}^{-S_i})^{-1}\right)' = f_\sigma(S_i)\,(1 - f(S_i))\,. \tag{4.20}$$

Подставив это выражение в (4.19), получим:

$$\Delta w_{ij} = \eta(d_i - y_i)f_\sigma(S_i)\,(1 - f(S_i))\,x_j = \eta(d_i - y_i)y_i(1 - y_i)x_j. \tag{4.21}$$

Таким образом, мы получили *итерационную формулу* для обучения персептрона

$$w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij}, \tag{4.22}$$

в которой

$$\Delta w_{ij} = \eta\delta_i x_j, \tag{4.23}$$

$$\delta_i = y_i(1 - y_i)(d_i - y_i). \tag{4.24}$$

Величину δ_i, введенную здесь с помощью формулы (4.24), в дальнейшем будем называть *нейронной ошибкой*. Алгоритм (4.22)–(4.24) называют *обобщенным дельта-правилом*. Его преимущество по сравнению с обычным дельта-правилом состоит в более быстрой сходимости и в возможности более точной обработки входных и выходных непрерывных сигналов, т. е. в расширении круга решаемых персептронами задач.

Итак, введение сигмоидной функции активации вместо функции-ступеньки и появление нового алгоритма обучения — обобщенного дельта-правила — расширили область применения персептрона. Теперь он может оперировать не только с бинарными (типа «нуль» и «единица»), но и с непрерывными (аналоговыми) выходными сигналами.

Контрольные вопросы и задания к § 4.5

1. Нарисуйте сигмоидную активационную функцию и напишите ее математическую формулу.
2. Чем сигмоидная функция активации лучше (или хуже) функции-ступеньки?
3. Напишите формулу для вычисления квадратичной ошибки персептрона. От каких величин она зависит?
4. Для чего нужен множитель 1/2 в формуле для квадратичной ошибки обучения персептрона? Что будет, если этот множитель не использовать?
5. В виде какой геометрической фигуры изображается квадратичная ошибка обучения персептрона?
6. Чем гиперпсевдопараболоид отличается от псевдопараболоида?
7. В чем суть метода градиентного спуска?
8. Попробуйте применить алгоритм метода градиентного спуска к задаче поиска точки минимума функции $y = x^2$.
9. Напишите формулы итерационного процесса, соответствующего обобщенному дельта-правилу.
10. Можно ли применять алгоритм обычного (необобщенного) дельта-правила для обучения персептрона с сигмоидными активационными функциями?
11. Можно ли применять обобщенное дельта-правило для обучения персептрона со ступенчатыми активационными функциями?
12. Какие преимущества и недостатки имеет обобщенное дельта-правило перед необобщенным?

4.6. ОГРАНИЧЕННОСТЬ ОДНОСЛОЙНОГО ПЕРСЕПТРОНА

Итак, ученым удалось обучить персептрон распознавать буквы алфавита. Это был колоссальный успех: *электронное устройство, созданное по образу и подобию человеческого мозга, обученное подобно человеку, успешно моделировало интеллектуальные функции человека.* Это был успех в познании самой природы человеческого мышления. Мозг начал раскрывать свои тайны. Появилась возможность исследовать мозг методами моделирования, не прибегая к сложнейшим антигуманным и мало что дающим натурным экспериментам. Это была сенсация, приковавшая к себе внимание мыслящих людей всего мира. Казалось, что ключ к интеллекту был найден и полное воспроизведение человеческого мозга и всех его функций — всего лишь вопрос времени. Писателям-фантастам, ученым, инженерам, бизнесменам, политикам виделись самые радужные перспективы практического применения идей искусственного интеллекта. Правительство Соединенных Штатов Америки выделило крупные субсидии на развитие нового перспективного научного направления.

Благодаря изобретению сигмоидных активационных функций и алгоритма градиентного спуска класс решаемых нейросетями задач расширялся. Делались попытки применения персептронов для решения задач прогнозирования, таких как предсказание погоды, курсов валют и акций. Персептроны пытались применять для анализа электрокардиограмм, для решения задач медицинской диагностики.

Но по мере расширения фронта научных исследований появлялись трудности. Неожиданно оказалось, что многие новые задачи персептрон решить не мог, потому что с ростом числа эпох ошибка обучения ε не стремилась к нулю. Если для одних задач кривая, изображающая зависимость ε от t, быстро приближалась к оси абсцисс (рис. 4.14, *а*), то для других задач (и их было подавляющее большинство!) погрешность ε обучения не удавалось снизить даже при большом количестве эпох t (рис. 4.14, *б*).

В качестве примера можно привести провал американского проекта создания системы противовоздушной обороны с использования персептрона. Ни огромные гранты научных исследований, ни успехи в создании быстродействующих компьютеров не помогли обучить персептрон решать задачи распознавания движущихся военных объектов на «свой» и «чужой». Причем эти новые важные задачи в плане математической постановки практически ничем не отличались от тех, с которыми персептрон успешно справлялся ранее. Время шло, итерации продолжались, а погрешность обучения не падала. Возникла необходимость объяснения парадоксов, глубокого анализа и создания теоретической базы нейроинформатики.

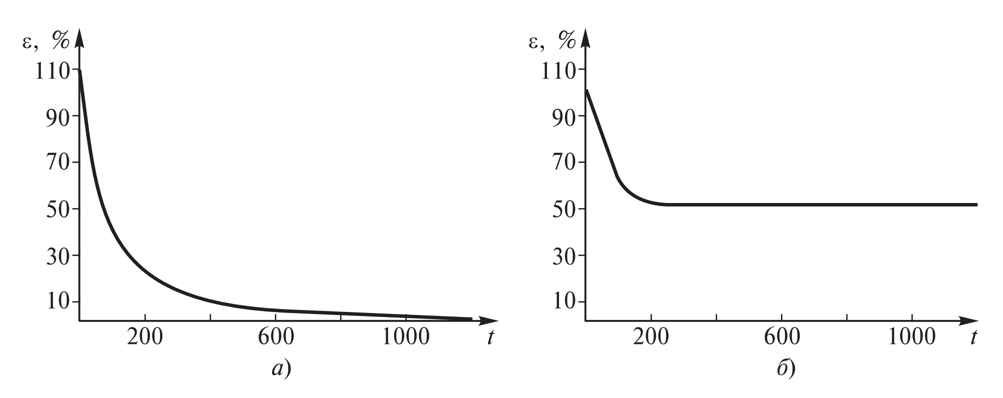


Рис. 4.14. Случай, когда ошибка обучения ε с ростом числа эпох t стремится к нулю (*а*), и случай, когда такого стремления нет (*б*)

Следующий период истории искусственного интеллекта начался с появления в 1969 г. книги двух известных американских математиков *М. Минского* и *С. Пейперта* «Персептроны» [34]. Авторы этой книги математически строго доказали, что использовавшиеся в то время однослойные персептроны в принципе не способны решать многие простые задачи. Одну из таких задач, вошедшую в историю нейроинформатики под названием проблемы «Исключающего ИЛИ», мы рассмотрим подробно.

«Исключающее ИЛИ» — это логическая функция двух аргументов, каждый из которых может иметь значение «истинно» либо «ложно». Сама она принимает значение «истинно», когда только один из аргументов имеет значение «истинно». Во всех остальных случаях эта функция принимает значение «ложно». Если закодировать значение «истинно» единицей, а значение «ложно» — нулем, то требуемое соответствие между аргументами x_1, x_2 и самой функцией y можно представить в виде табл. 4.3, называемой таблицей истинности логической функции.

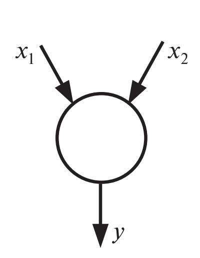


Рис. 4.15. Однонейронный персептрон с двумя входами и одним выходом

Логическая функция «Исключающее ИЛИ» может быть выражена через функции логического умножения «И» («AND»), логического сложения «ИЛИ» («OR») и логического отрицания «НЕТ» («NOT») с помощью логической формулы

$$y = (x_1 \text{AND NOT } x_2) \text{ OR } (x_2 \text{AND NOT } x_1). \quad (4.25)$$

Задача состоит в том, чтобы научиться моделировать функцию «Исключающее ИЛИ» с помощью однонейронного персептрона с двумя входами x_1 и x_2 и одним выходом y (рис. 4.15).

Таблица 4.3

Таблица истинности логической функции «Исключающее ИЛИ»

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Таблица 4.4

Таблица истинности логической функции «Исключающее ИЛИ», дополненная точками *A, B, C, D*

Точки	x_1	x_2	y
A	0	0	0
B	0	1	1
C	1	0	1
D	1	1	0

М. Минский и С. Пейперт в своей книге [34] предложили геометрическую интерпретацию к проблеме «Исключающего ИЛИ», состоящую в следующем. Они предложили изобразить на координатной плоскости x_1, x_2 все возможные комбинации входных сигналов в виде четырех точек: *A, B, C, D,* как показано на рис. 4.16. Точка A имеет координаты $x_1 = 0$, $x_2 = 0$; точка B имеет координаты $x_1 = 0$, $x_2 = 1$ и т. д., согласно табл. 4.4.

Тогда в точке A выход персептрона y должен быть равен нулю, в точке B — единице, в точке C — единице и в точке D — нулю.

В соответствии с формулами (4.1)–(4.2) однонейронный персептрон (см. рис. 4.15) осуществляет преобразование

$$S = w_1x_1 + w_2x_2; \qquad (4.26)$$

$$y = \begin{cases} 1, & \text{если } S \geqslant \theta; \\ 0, & \text{если } S < \theta. \end{cases} \qquad (4.27)$$

Рассмотрим случай, когда $S = \theta$. Это значит, что, согласно (4.26), выполняется равенство:

$$w_1x_1 + w_2x_2 = \theta. \qquad (4.28)$$

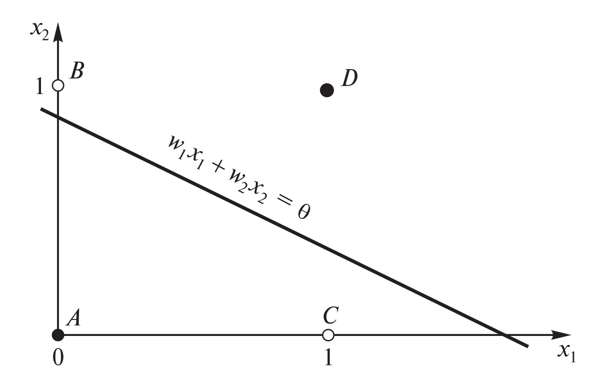


Рис. 4.16. Геометрическая интерпретация к объяснению проблемы «Исключающего ИЛИ»

Если в этом уравнении величины x_1 и x_2 считать переменными, а θ, w_1 и w_2 — константами, то на координатной плоскости x_1, x_2 рассматриваемое уравнение изобразится в виде прямой линии, положение и наклон которой определяются значениями коэффициентов w_1, w_2 и порога θ. Для всех точек плоскости x_1, x_2, лежащих на этой линии, выполняется равенство $S = \theta$, и поэтому, согласно формуле (4.27), выходной сигнал персептрона равен единице. Для точек, лежащих выше указанной линии, сумма $w_1x_1 + w_2x_2$ больше θ, и поэтому, согласно формулам (4.26)–(4.27), выходной сигнал персептрона также равен единице. Для точек же, лежащих ниже этой линии, сумма $w_1x_1 + w_2x_2$ меньше θ, и выходной сигнал персептрона равен нулю. Поэтому линию, изображающую уравнение (4.28), называют ***пороговой прямой***.

А теперь посмотрим на таблицу истинности функции «Исключающее ИЛИ» (см. табл. 4.4). Согласно этой таблице, в точках A и D выход персептрона должен быть нулевым, а в точках B и C — единичным. Но для этого надо расположить пороговую прямую так, чтобы точки A и D лежали ниже этой линии, а точки B и C — выше, что невозможно. Это значит, что, сколько бы персептрон ни обучали, какие бы значения ни придавали его синаптическим весам и порогу, персептрон в принципе не способен воспроизвести соотношение между входами и выходом, требуемое таблицей истинности функции «Исключающее ИЛИ».

Помимо проблемы «Исключающего ИЛИ», в упомянутой выше книге М. Минский и С. Пейперт привели ряд других задач, в которых точки, изображающие входные сигналы, не могут быть разделены пороговой прямой (в многомерных случаях — плоскостью, гиперплоскостью). Такие задачи получили название ***линейно неразделимых***.

После выхода в свет книги М. Минского и С. Пейперта «Персептроны» всем стало ясно, что предпринимавшиеся в то время попытки обучать персептроны решению многих задач, которые, как оказалось, относятся к классу линейно неразделимых, с самого начала были обречены на провал. Это была пустая трата времени, сил и финансовых ресурсов. Успешность же обучения персептрона распознаванию букв латинского алфавита — это счастливая случайность. Она объясняется тем, что задача оказалась линейно разделимой, что в жизни встречается крайне редко.

Итак, доказано, что однонейронный персептрон не позволяет моделировать логическую функцию «Исключающее ИЛИ» и решать другие линейно неразделимые задачи.

Контрольные вопросы и задания к § 4.6

1. Нарисуйте таблицы истинности логических функций «И», «ИЛИ», «Исключающее ИЛИ».

2. Дайте определение пороговой прямой.
3. Перерисуйте рис. 4.16 и начертите на нем пороговую прямую так, чтобы однонейронный персептрон, параметры которого соответствуют нарисованной вами пороговой прямой, моделировал:
   – логическую функцию «И»,
   – логическую функцию «ИЛИ».
4. Пользуясь своим рисунком, объясните, почему однонейронный персептрон не может моделировать функцию «Исключающее ИЛИ».
5. Чем можно объяснить, что однослойный персептрон, несмотря на обнаружившийся в этом параграфе недостаток, все-таки смог научиться распознавать буквы латинского алфавита?
6. Дайте определение линейно неразделимых задач.
7. Подумайте над тем, как заставить персептрон решать линейно неразделимые задачи.

4.7. ПЕРСЕПТРОН СО СКРЫТЫМ СЛОЕМ НЕЙРОНОВ

Появление книги М. Минского и С. Пейперта «Персептроны» вызвало шок в научном мире. Строгие математические доказательства М. Минского и С. Пейперта были неуязвимы. Всеобщий энтузиазм сменился не менее всеобщим пессимизмом. В газетах стали появляться критические статьи с сообщениями о том, что ученые мужи в своих исследованиях зашли в тупик, впустую израсходовав деньги налогоплательщиков. Правительство США немедленно прекратило финансирование нейропроектов и приступило к поискам виновных в растрате государственных денег. Бизнесмены, потерявшие надежду вернуть вложенные капиталы, отвернулись от ученых, и нейрокибернетика была предана забвению, длившемуся более 20 лет.

Тем не менее работы в области нейросетевых и нейрокомпьютерных технологий продолжались отдельными энтузиастами. Работы продолжались и в засекреченных научно-исследовательских институтах Советского Союза, отделенного в то время от Запада «железным занавесом». Не имея информации о настроениях зарубежных коллег, советские ученые спокойно продолжали заниматься захватившей их умы темой и к началу 80-х гг. удивили мир появлением не только нейрокомпьютерных систем распознавания «свой-чужой», но и нейрокомпьютеров, управляющих полетами ракет и самолетов. Советские системы управления, в отличие от американских, стойко переносили довольно серьезные повреждения, продолжая работать в сложных условиях, что было особенно важно для объектов военного назначения. Выявилось еще одно, очень важное для военных свойство нейрокомпьютеров, унаследованное ими от мозга, — свойство *живучести*.

Советским ученым *С. О. Мкртчяном* была издана книга «Нейроны и нейронные сети. Введение в теорию формальных нейронов» [35],

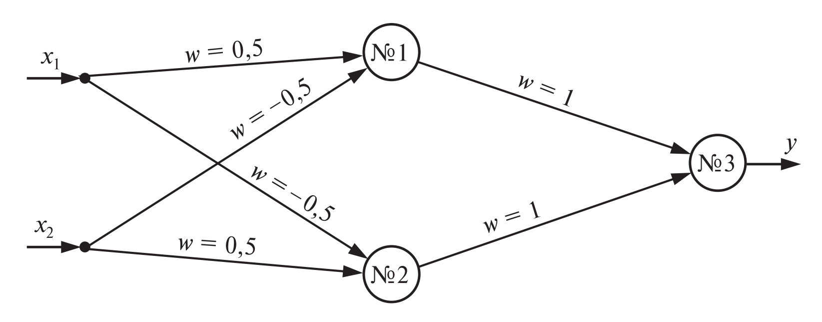


Рис. 4.17. Нейронная сеть, моделирующая функцию «Исключающее ИЛИ»

в которой он показал, что с помощью многослойных персептронов может быть смоделирована любая логическая функция, если только известна ее логическая формула. Более того, им был разработан специальный математический аппарат, позволяющий конструировать такие персептроны. Оказалось, что проблема «Исключающего ИЛИ», явившаяся камнем преткновения для однонейронного персептрона, может быть разрешена с помощью нейронной сети, состоящей из трех нейронов — трехнейронного персептрона, изображенного на рис. 4.17.

Работа этого персептрона происходит по следующему алгоритму.

Нейрон № 1:

$$S_1 = 0,5x_1 + (-0,5)x_2;$$

$$y_1 = \begin{cases} 1, & \text{если } S_1 \geqslant \theta; \\ 0, & \text{если } S_1 < \theta. \end{cases}$$

Нейрон № 2:

$$S_2 = (-0,5)x_1 + 0,5x_2;$$

$$y_2 = \begin{cases} 1, & \text{если } S_2 \geqslant \theta; \\ 0, & \text{если } S_2 < \theta. \end{cases}$$

Нейрон № 3:

$$S_3 = 1 \cdot y_1 + 1 \cdot y_2;$$

$$y_3 = \begin{cases} 1, & \text{если } S_3 \geqslant \theta; \\ 0, & \text{если } S_3 < \theta. \end{cases}$$

Задавшись значением порога $\theta = 0,5$ и заполнив с помощью этих формул табл. 4.5, легко убедиться, что трехнейронный персептрон успешно моделирует функцию «Исключающее ИЛИ».

Впоследствии было показано, что и другие линейно неразделимые задачи, приведенные в книге М. Минского и С. Пейперта, могут быть решены с помощью нейросетей, содержащих один или несколько *скрытых* нейронных слоев, т. е. слоев нейронов, расположенных между входным и выходным слоями.

Многие исследователи понимали, что нужно создавать нейросети более сложной архитектуры, содержащие скрытые слои нейронов, но не представляли, как такие сети обучать. Правила Хебба и дель-

Таблица 4.5

Процесс формирования сигналов в трехнейронном персептроне

x_1	x_2	S_1	S_2	y_1	y_2	S_3	y_3	y
0	0	0	0	0	0	0	0	0
0	1	–0,5	0,5	0	1	1	1	1
1	0	0,5	–0,5	1	0	1	1	1
1	1	0	0	0	0	0	0	0

та-правило годились только для корректировки синаптических весов нейронов выходного слоя, тогда как вопрос о настройке параметров скрытых нейронных слоев оставался открытым.

Контрольные вопросы и задания к § 4.7

1. Нарисуйте персептрон, моделирующий функцию «Исключающее ИЛИ».
2. С помощью формул, описывающих работу математического нейрона, убедитесь, что нарисованный вами персептрон действительно моделирует функцию «Исключающее ИЛИ».
3. Попробуйте изобразить другой персептрон (другой структуры), тоже способный моделировать логическую функцию «Исключающее ИЛИ».
4. Почему не удается применять известные вам алгоритмы обучения (правила Хебба, дельта-правило, обобщенное дельта-правило) для обучения персептронов, моделирующих функцию «Исключающее ИЛИ»?
5. Попытайтесь придумать алгоритм обучения персептрона, содержащего один скрытый слой.

4.8. МНОГОСЛОЙНЫЙ ПЕРСЕПТРОН И АЛГОРИТМ ЕГО ОБУЧЕНИЯ

Эффективный алгоритм обучения многослойных персептронов, открывший путь их широкому практическому применению, стал известен только в 1986 г. благодаря публикациям *Д. Румельхарта, Г. Хилтона* и *Р. Вильямса* [103]. Идея этого алгоритма заключается в том, что ошибки нейронов выходного слоя $\varepsilon_i = d_i - y_i$ используются для вычисления ошибок нейронов, расположенных в скрытых слоях. Значения ошибок как бы распространяются от выходного слоя нейронов вовнутрь сети от последующих нейронных слоев к предыдущим. Отсюда название метода: *алгоритм обратного распространения ошибки* (*back propagation*).



Рис. 4.18. Профессор А. И. Галушкин — автор первых алгоритмов обучения многослойных нейронных сетей персептронного типа

Интересно отметить, что алгоритм обучения многослойных нейронных сетей, как и многие другие выдающиеся научные открытия, на самом деле имеет несколько авторов. Так, впоследствии выяснилось, что алгоритм обратного распространения ошибки был предложен на один год ранее в работах *А. Паркера* и *А. Ле-Кана,* изданных независимо одна от другой. Кроме того, оказалось, что еще в 1974 г. этот простой и изящный алгоритм был защищен *П. Дж. Вербосом* [106] в его докторской диссертации. Более того, идеи и алгоритмы обучения многослойных структур персептронного типа можно найти в еще более ранних публикациях советских ученых: *А. И. Галушкина* (рис. 4.18), *В. А. Ванюшина*, *Л. С. Зака*, *Б. П. Тюхова* [4, 6–10], относящихся к 1970–1974 гг. Последнее обстоятельство позволяет сделать заключение о приоритете Советского Союза в этом воистину колоссальном научном открытии, которое вывело нейроинформатику из тупика и превратило ее в самую прикладную научную отрасль.

Рассмотрим идею алгоритма обратного распространения ошибки, попытавшись обобщить дельта-правило на случай обучения *двухслойного* персептрона, имеющего N входов, I выходов и *скрытый* слой из J нейронов (рис. 4.19). На самом деле этот персептрон имеет три слоя, однако в литературе его называют двухслойным, поскольку нейроны входного слоя имеют всего один вход, не имеют синаптических весов и не выполняют суммирования входных сигналов, а лишь передают один-единственный входной сигнал нейронам следующего слоя.

Алгоритм корректировки синаптических весов нейронов выходного слоя оставим таким же, как для однослойного персептрона (см. обоб-

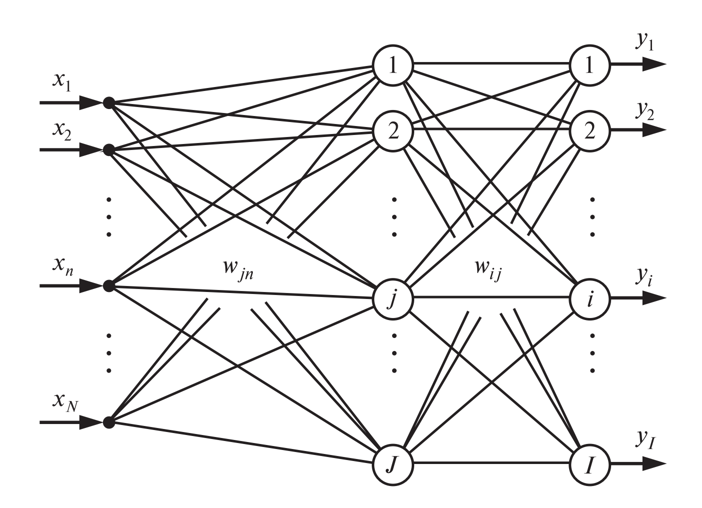


Рис. 4.19. Двухслойный персептрон с N входами, I выходами и скрытым слоем из J нейронов

щенное дельта-правило формулы (4.22)–(4.24)), заменив в них x_j на y_j [1]:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}; \tag{4.29}$$

$$\Delta w_{ij} = \eta \delta_i y_j; \tag{4.30}$$

$$\delta_i = y_i(1 - y_i)(d_i - y_i). \tag{4.31}$$

Синаптические веса нейронов скрытого слоя попытаемся корректировать с помощью все тех же формул (4.22)–(4.24), в которых индекс i заменим на j, а индекс j заменим на индекс n:

$$w_{jn}(t+1) = w_{jn}(t) + \Delta w_{jn}; \tag{4.32}$$

$$\Delta w_{jn} = \eta \delta_j x_n; \tag{4.33}$$

$$\delta_j = y_j(1 - y_j)(d_j - y_j). \tag{4.34}$$

При использовании этих формул возникает вопрос о вычислении нейронной ошибки $(d_j - y_j)$, которая для скрытого слоя неизвестна. Идея авторов рассматриваемого алгоритма состояла в том, чтобы в качестве этой ошибки использовать суммарные нейронные ошибки с выходного слоя, помноженные на силы соответствующих синаптических связей, т. е.

$$(d_j - y_j) = \sum_{i=1}^{I} \delta_i w_{ij}. \tag{4.35}$$

[1] Причины этой замены становятся понятными, если проследить пути прохождения сигналов, сравнивая рис. 4.12 и рис. 4.19: если в однослойном персептроне (рис. 4.12) на вход нейронов выходного слоя подавались входные сигналы x_j, то в двухслойном персептроне (рис. 4.19) на вход нейронов выходного слоя подаются выходные сигналы нейронов с предыдущего (скрытого) слоя y_j.

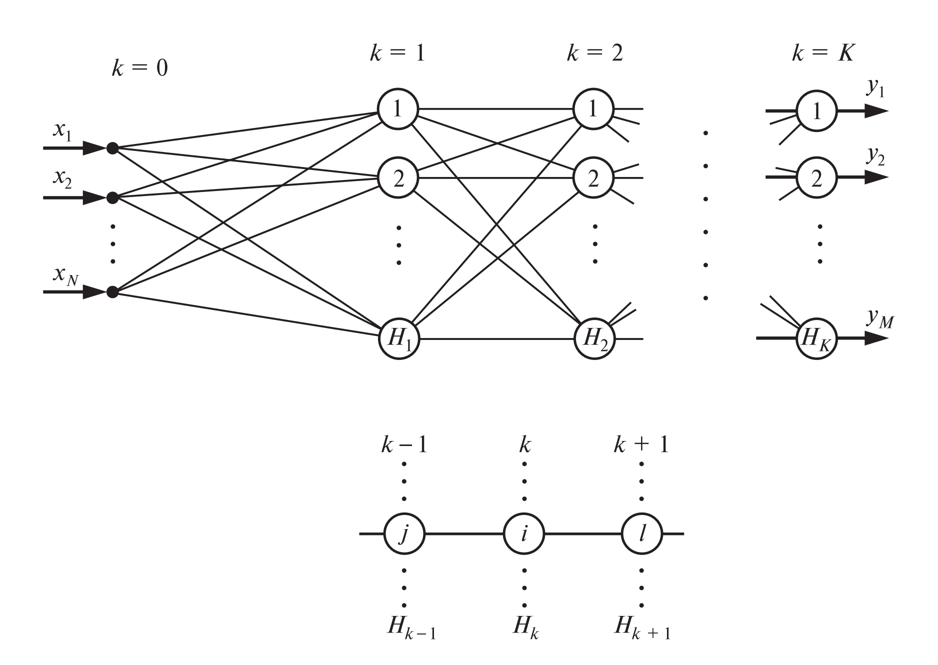


Рис. 4.20. Многослойный персептрон

Итак, для скрытого слоя окончательно имеем:

$$\Delta w_{jn} = \eta \delta_j x_n; \tag{4.36}$$

$$\delta_j = y_j(1 - y_j) \sum_{i=1}^{I} \delta_i w_{ij}. \tag{4.37}$$

Воспользовавшись этой идеей, несложно расписать алгоритм обратного распространения ошибки для обучения персептрона, имеющего произвольное количество скрытых слоев. Однако прежде отметим, что мы будем использовать нейроны, имеющие сигмоидную активационную функцию (4.13) и выполняющие операцию суммирования по формуле (4.6). Согласно этой формуле, каждый i-й нейрон вычисляет сумму

$$S_i = \sum_{j=0}^{J} w_{ij} x_j,$$

в которой w_{i0} — вес дополнительного входа i-го нейрона, имитирующий его смещение b_i, а $x_0 = 1$ — величина сигнала дополнительного входа.

Алгоритм обратного распространения ошибки распишем для многослойного персептрона, имеющего входной слой $k = 0$, несколько скрытых слоев $k = 1, 2, \ldots, K - 1$ и выходной слой $k = K$ (рис. 4.20).

Нейроны входного слоя не выполняют математических преобразований, а лишь передают входные сигналы нейронам первого слоя. Будем полагать, что каждый k-й слой содержит H_k нейронов. Таким образом,

персептрон имеет $N = H_0$ входов и $M = H_K$ выходов. В алгоритме будем использовать следующие обозначения: i — порядковый номер нейрона k-го слоя; j — порядковый номер нейрона $(k-1)$-го слоя; l — порядковый номер нейрона $(k+1)$-го слоя (см. рис. 4.20, *внизу*).

Шаг 1. Инициализация синаптических весов и смещений.

В циклах по $k = 1, 2, \ldots, K$; $i = 1, 2, \ldots, H_k$; $j = 0, 1, 2, \ldots, H_{k-1}$ датчик случайных чисел присваивает синаптическим весам и смещениям $w_{ij}^{(k)}$ малые величины, например из интервала от -1 до 1.

Шаг 2. Открытие цикла по $q = 1, 2, \ldots, Q$. Представление из обучающего множества примеров очередного входного вектора $X_q = (x_1, x_2, \ldots, x_N)_q$ и соответствующего ему желаемого выходного вектора $D_q = (d_1, d_2, \ldots, d_M)_q$, где q — номер примера в обучающем множестве.

Шаг 3. Прямой проход.

В циклах по $k = 1, 2, \ldots, K$; $i = 1, 2, \ldots, H_k$ вычисляются выходные сигналы i–го нейрона в k-м слое

$$y_i^{(k)} = f_\sigma\left(\sum_{j=0}^{H_{k-1}} w_{ij}^{(k)} y_j^{(k-1)}\right), \tag{4.38}$$

где $y_j^{(0)} = x_j$; $x_0 = 1$; $y_0^{(k-1)} = 1$; $y_i = y_i^{(K)}$ — выходные сигналы персептрона.

Шаг 4. Обратный проход.

В циклах по $k = K, K-1, \ldots, 1$; $i = 1, 2, \ldots, H_k$; $j = 0, 1, 2, \ldots, H_{k-1}$ вычисляются синаптические веса на новой эпохе

$$w_{ij}^{(k)}(t+1) = w_{ij}^{(k)}(t) + \Delta w_{ij}^{(k)} \quad , \tag{4.39}$$

где

$$\Delta w_{ij}^{(k)} = \eta \delta_i^{(k)} y_j^{(k-1)}, \tag{4.40}$$

причем для выходного слоя $k = K$ согласно (4.30)

$$\delta_i^{(K)} = y_i(1 - y_i)(d_i - y_i),$$

а для всех других скрытых слоев согласно (4.37)

$$\delta_i^{(k)} = y_i^{(k)}(1 - y_i^{(k)}) \sum_{l=1}^{H_{k+1}} \delta_l^{(k+1)} w_{li}^{(k+1)}.$$

Шаг 5. Закрытие цикла по q.

Шаг 6. Повторение шагов 2–5 необходимое количество раз.

Векторы обучающих примеров X_q и D_q на *шаге 2* алгоритма обычно представляются последовательно от первого до последнего, т. е. $q = 1, 2, \ldots, Q$, где Q — общее количество примеров. Например, в случае распознавания букв русского алфавита $Q = 33$. После того как для каждого обучающего примера будут скорректированы весовые

коэффициенты персептрона, т. е. *шаги 2—4* будут повторены 33 раза, *на шаге 6* алгоритма вычисляется среднеквадратичная ошибка, усредненная по всем обучающим примерам:

$$\varepsilon = \sqrt{\frac{1}{QM}\sum_{q=1}^{Q}\sum_{i=1}^{M}\left((d_i - y_i)_q^2\right)}. \qquad (4.41)$$

Помимо среднеквадратичной ошибки, может быть также оценена максимальная разность между желаемым и прогнозным (то, что вычислил персептрон) выходами персептрона:

$$\varepsilon = \max\left((|d_i - y_i|)_q\right); \quad i = 1, 2, \ldots, M; \quad q = 1, 2, \ldots, Q. \qquad (4.42)$$

Итерационный процесс, задаваемый *шагом 6*, заканчивается после того, как ошибка ε, вычисляемая по формулам (4.40) или (4.42), достигнет заданной величины либо когда будет достигнуто предельное количество эпох обучения. В результате персептрон обучится выполнять нужное отображение любого входного вектора $\boldsymbol{X}_q$ на выходной вектор $\boldsymbol{Y}_q$, отличающийся от желаемого вектора $\boldsymbol{D}_q$ на некоторую малую величину.

Теперь представим, что на входное табло фотоэлементов попала карточка с какой-либо буквой, выполненной другим шрифтом. Фотоэлементы сформируют входной вектор $\boldsymbol{X}$, не совпадающий ни с одним из векторов множества обучающих примеров. Если шрифт, которым выполнена входная буква, не слишком отличается от шрифта примеров обучающего множества, а персептрон хорошо спроектирован и обучен, то он вычислит вектор $\boldsymbol{Y}$, в котором выход нейрона, соответствующего представленной на вход букве, будет иметь максимальное значение. Таким образом, персептрон, несмотря на помехи и искажения входного образа, выдаст правильное заключение о его принадлежности к тому или иному классу. Свойство персептрона «узнавать» образы, которых не было во множестве обучающих примеров, называется свойством *обобщения*. Как уже было отмечено ранее, это свойство, которое персептрон унаследовал от своего прототипа — мозга, имеет очень важное значение для его практических приложений.

Подведем итоги изучения алгоритмов обучения нейронных сетей.

Первым алгоритмом были правила Хебба, предназначенные для обучения однослойного персептрона с нейронами, имеющими ступенчатые активационные функции. Затем было введено понятие нейронной ошибки как разницы между требуемым выходом нейрона d_i и его реальным значением y_i. В результате алгоритм обучения персептрона с помощью правил Хебба был обобщен в виде алгоритма дельта-правила. В итерационных формулах алгоритма дельта-правила появился коэффициент скорости обучения η, позволяющий влиять на

величину итерационного шага. Затем была предложена сигмоидная активационная функция и было введено понятие квадратичной ошибки обучения персептрона. В результате появилось обобщенное дельта-правило, реализующее метод градиентного спуска и позволяющее работать не только с бинарными, но и с непрерывными сигналами. Алгоритм обратного распространения ошибки является следующим обобщением обобщенного дельта-правила и позволяет обучать не только однослойные, но и многослойные персептроны.

Контрольные вопросы и задания к § 4.8

1. Объясните, в чем состоит идея алгоритма обратного распространения ошибки. Отражает ли название алгоритма его идею?
2. Какую роль в методе обратного распространения ошибки выполняет коэффициент скорости обучения η?
3. Попробуйте запрограммировать алгоритм обратного распространения ошибки на каком-либо алгоритмическом языке.
4. Сколько алгоритмов обучения нейронных сетей вам известно? Назовите их и охарактеризуйте их возможности.
5. Годится ли алгоритм обратного распространения ошибки для обучения персептрона со ступенчатыми активационными функциями?
6. Годятся ли правила Хебба для обучения персептрона с нейронами, имеющими сигмоидные функции активации?
7. Годится ли дельта-правило для обучения персептрона с нейронами, имеющими сигмоидные функции активации?
8. Назовите преимущества и недостатки алгоритма обратного распространения ошибки по сравнению со всеми изученными ранее методами обучения нейронных сетей.
9. Зайдите на сайт **www.LbAi.ru** и выполните лабораторную работу № 5 — обучите персептрон моделировать функции «И», «ИЛИ», «Исключающее ИЛИ». В случае затруднений или заинтересованности обратитесь к учебно-методическим пособиям [73, 88].

Глава 5
ВОЗМОЖНОСТИ И СФЕРЫ ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ

5.1. ВОЗМОЖНОСТИ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

С появлением алгоритма обратного распространения ошибки начался период широкого практического применения нейросетевых технологий для решения самых разнообразных задач. С помощью многослойного персептрона стало возможным строить математические модели, выполняющие сложные многомерные отображения входного вектора параметров $\boldsymbol{X}$ на выходной вектор $\boldsymbol{Y}$.

Задачи подобного рода часто встречаются в самых разнообразных, казалось бы, не имеющих ничего общего областях, таких как промышленность, экономика, бизнес, финансы, политология, социология, психология, экология, медицина и т. д.

Практически в каждой проблеме, решаемой прикладными науками, требуется строить модели явлений, процессов, объектов, т. е. выявлять и математически описывать зависимости одних комплексов параметров от других. Требуется строить математические функции, которые можно использовать для более глубокого анализа моделируемых объектов, например найти оптимальное сочетание управляющих параметров, обеспечивающих максимум некоторой целевой функции, такой как рентабельность, прибыльность, прочность, температура, скорость, высота и т. д. Или выполнить прогнозирование, т. е. предсказать, как будут развиваться события в зависимости от того или иного воздействия на моделируемый объект, и как повлиять на эти события путем выбора нужного воздействия.

При обучении в вузе, да и просто в жизни вы не раз сталкивались с методом математического моделирования и наверняка оценили его эффективность. Например, в школе вы решали задачу о движении тела, брошенного под углом к горизонту. Имея математическую модель этого явления — уравнения, связывающие угол бросания, начальную скорость, высоту подъема и дальность полета, исследуя эту математическую модель, решая уравнения, просчитывая разные варианты выбора исходных параметров, вы легко определили оптимальный угол бросания, обеспечивающий максимальную дальность полета тела. Для этого вам не надо было ставить никаких натурных экспериментов, не потребовалось выходить во двор и бросать камни.

Говоря о важности метода математического моделирования в нашей жизни, отметим, что в настоящее время он является одним из самых эффективных методов получения научных знаний. Методом математического моделирования рассчитываются, проектируются, оптимизируются новые инженерные и строительные конструкции, делается прогноз погоды, предсказываются стихийные бедствия, выполняются экономические прогнозы, на основе которых строится экономическая политика отдельных фирм и целых государств.

Новые научные знания, полученные методом математического моделирования, не раз оказывали решающее влияние на формирование нашей цивилизации. Несколько таких примеров приведены в книге [68].

1. Оптимизация формы крыла самолета методом математического моделирования, выполненная Н. Е. Жуковским, С. А. Чаплыгиным и их учениками, положила начало успешного развития авиации.

2. Изобретение артиллерийских снарядов кумулятивного действия и противотанковой гранаты, спроектированных методом математического моделирования академиком М. А. Лаврентьевым с учениками [32], способствовало перелому в ходе Второй мировой войны.

3. Успехи в развитии методов решения краевых задач и математическое моделирование напряженно-деформированного состояния ракетных двигателей [77] способствовали достижению опережающих темпов освоения космического пространства Советским Союзом в 1960-х и начале 1970-х гг.

4. Благодаря методу математического моделирования было открыто явление *ядерной зимы* — глобальное понижение температуры поверхности планеты, вызванное массовыми ядерными взрывами. Никто и никогда это явление в действительности не наблюдал. Оно было открыто [1, 99] на экране компьютера в результате вычислительных экспериментов над математическими компьютерными моделями. И это открытие радикальным образом повлияло на государственную политику великих держав. Стало ясно, что победителей в ядерной войне не будет. Бессмысленная гонка вооружений была прекращена. Глобальный мир на планете был сохранен.

До появления нейронных сетей и нейрокомпьютеров математические модели традиционно строились на основе фундаментальных законов природы, таких как законы сохранения массы, энергии, количества движения и др. Эти законы записывались в виде алгебраических либо дифференциальных и интегральных уравнений, к которым добавлялись уравнения, отражающие закономерности конкретных предметных областей. Для получения результата приходилось разрабатывать и применять алгоритмы совместного решения всех этих уравнений, составляющих математическую модель исследуемой предметной области.

Нейроинформационные технологии открыли иной подход к самой методике построения компьютерных математических моделей. Теперь, не задумываясь над законами физики, химии, биологии, медицины, общественного развития и т. д., исходя из одного только эмпирического опыта, представленного обучающими примерами, можно строить компьютерные математические модели, которые сами извлекают эти законы и позволяют их эффективно использовать для решения широкого круга практических задач. Появился новый инструмент извлечения знаний из данных, позволяющий заново открывать фундаментальные законы природы, выявлять ранее неизвестные зависимости и использовать их для решения конкретных практических задач. Нейронные сети стали одним из наиболее эффективных инструментов интеллектуального анализа данных.

Особенно эффективен этот новый инструмент оказался при построении математических моделей и интеллектуального анализа данных плохо формализуемых предметных областей, таких как, например, медицина.

5.2. ДИАГНОСТИКА ЗАБОЛЕВАНИЙ ЧЕЛОВЕКА

Первые попытки создания нейросетевых медицинских диагностических систем относятся к 40–50 гг. XX в., т. е. сразу после появления основополагающих работ У. Мак-Калока, У. Питтса, Ф. Розенблатта, заложивших теоретический фундамент нейросетевых и нейрокомпьютерных технологий. Однако практическое применение такие системы начали находить только в 1980–1990 гг. В настоящее время в средствах информации и научной литературе имеется множество сообщений об удачном опыте применения нейронных сетей для постановки диагнозов заболеваний в различных областях медицины. Рассмотрим, как строятся и обучаются такие сети.

Проведем наблюдение за тем, как врач ставит диагноз болезни пациента. Прежде всего он выясняет и записывает имя, возраст, пол, место работы, затем, как правило, измеряет давление, проводит внешний осмотр, выслушивает жалобы больного, знакомится с историей его болезни, результатами анализов, данными рентгенографии, изучает электрокардиограмму. В результате у врача накапливается от 20 до 100 и более параметров, характеризующих пациента и состояние его здоровья. Это и есть исходные параметры, обработав которые с помощью своих медицинских знаний и опыта, врач делает заключение о заболевании пациента — ставит диагноз заболевания.

Задавшись целью построить нейросетевую математическую модель врача, мы прежде всего должны определиться с входным вектором $\boldsymbol{X}$ и выходным вектором $\boldsymbol{D}$, задав их размерности, и условиться о содержимом каждого компонента. В векторе $\boldsymbol{X}$ логично предусмотреть

параметры, которые врач выясняет у больного. Например, в качестве компоненты x_1 можно задать дату рождения, в качестве x_2 — закодировать пол (например, с помощью нуля и единицы), в качестве x_3 — вес больного, x_4 — артериальное давление, x_5 — температуру тела и т. д. Нелишне учесть также цвет глаз, цвет волос и другие данные, определяющие особенности организма и, следовательно, влияющие на вероятность возникновения тех или иных заболеваний.

В выходном векторе $\boldsymbol{D}$ следует закодировать все возможные диагнозы заболеваний, которые способен обнаружить врач.

Размерность вектора $\boldsymbol{D}$ можно существенно снизить, если моделировать врача, специализирующегося в узкой области медицины. Так, если мы выбрали врача-кардиолога, то в векторе $\boldsymbol{D}$ следует кодировать только заболевания сердечно-сосудистой системы. Например, можно принять $d_1 = 1$, если у больного инфаркт, и $d_1 = 0$, если инфаркта нет. Аналогично с помощью d_2 можно закодировать наличие или отсутствие ишемической болезни сердца (ИБС),..., d_M — стенокардии.

Таким образом, выходной вектор персептрона $\boldsymbol{D}$ будет состоять из множества нулей и одной или нескольких единиц (если болезней несколько). Однако, как показывает опыт [69, 108], медицинские диагнозы лучше кодировать по пяти-, десяти- или стобалльной шкале. Тогда на этапе подготовки обучающего множества примеров с помощью баллов можно будет учитывать степень уверенности врача в правильности его диагноза (или степень развитости заболевания). Например, если выходные параметры имеют значения: $d_1 = 0$, $d_2 = 100$ и $d_M = 75$, то это означает, что, по мнению врача, у пациента нет инфаркта миокарда, есть ишемическая болезнь сердца, и есть стенокардия, однако в последнем диагнозе он уверен не на все 100%, а только на 75% (или степень развитости оценивается в 75%).

Далее следует подготовить множество обучающих примеров. Мы воздержимся от советов по организации совместного труда врача и программиста, в результате которого будет создано необходимое количество обучающих примеров. Отметим только, что качество нейросетевой диагностической системы напрямую зависит от квалификации практикующего врача, на примерах работы которого она обучалась. Дело в том, что нейронная сеть наследует от врача не только его знания, но и пробелы в его медицинском образовании. Понятно, что она будет допускать те же врачебные ошибки, которые допускает врач. Поэтому для обеспечения высокого качества диагностики нейронную сеть следует обучать на примерах работы высококвалифицированного врача или даже на результатах работы врачебного консилиума. А если к работе по обучению нейронной сети привлечь еще и патологоанатома, исключающего ошибки врачебной диагностики, то будут все основания надеяться, что обученная таким способом нейронная сеть по качеству выставляемых диагнозов превзойдет врачей-людей.

В этом случае нейронная сеть может обнаружить и заложить в модель такие закономерности человеческого организма, которые современной медицине вообще неизвестны.

Отметим также, что при создании множества обучающих примеров может использоваться метод анкетирования: врач, выслушивая жалобы пациента, предварительно заполняет анкету, а программисты на основании анкетных данных формируют обучающие векторы $\boldsymbol{X_q}$ и $\boldsymbol{D_q}$.

Параметры входного вектора $\boldsymbol{X_q}$ могут поступать с какого-либо медицинского измерительного прибора: маммографа, электрокардиографа, импедансометра и др. Может также применяться смешанный способ сбора исходных данных: часть параметров поступают с медицинских приборов, а часть — с анкет.

В результате совместной работы коллектива специалистов-медиков и программистов будет накоплено множество обучающих примеров, состоящее из множества пар векторов $\boldsymbol{X_q}$ и $\boldsymbol{D_q}$ ($\boldsymbol{q} = 1, 2, \ldots, Q$). Теперь задача состоит в том, чтобы спроектировать персептрон и путем обучения передать ему знания и опыт, содержащиеся во множестве обучающих примеров. Вопросы проектирования персептронов, т. е. подбора количества скрытых слоев, количества содержащихся в них нейронов и типов активационных функций, рассматриваются далее в гл. 6, поэтому сейчас мы этим заниматься не будем (рис. 5.1).

В результате персептрон должен научиться отображать любой вектор обучающего множества $\boldsymbol{X_q}$ на вектор $\boldsymbol{Y_q}$, совпадающий (либо почти совпадающий) с вектором $\boldsymbol{D_q}$. Кроме того, при появлении нового пациента, характеризующегося новым входным вектором $\boldsymbol{X}_{Q+1}$, персептрон должен вычислить для него новый вектор

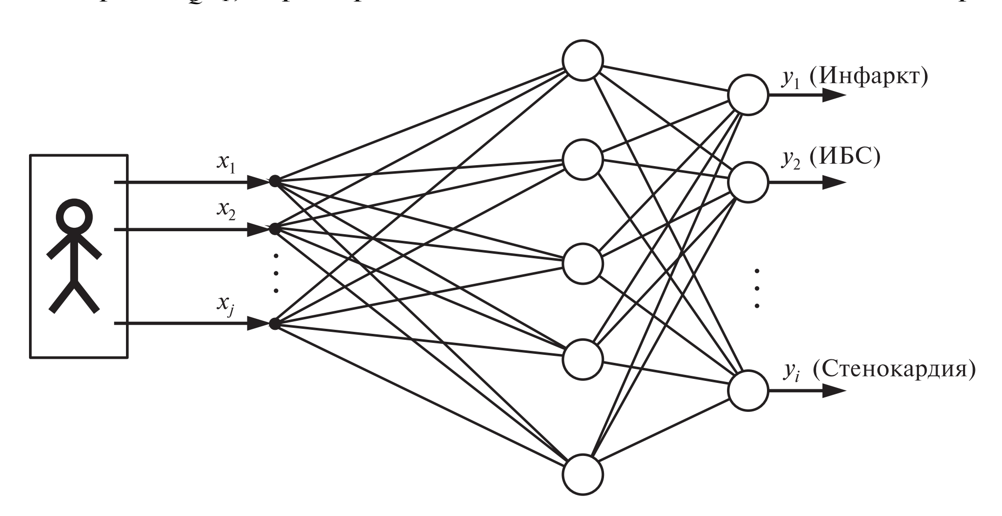


Рис. 5.1. Принципиальная схема нейросетевой системы медицинской диагностики

$\boldsymbol{Y}_{Q+1}$, содержащий правильный диагноз, поставленный персептроном уже без помощи врача. Другими словами, персептрон должен уметь ***обобщать*** переданный ему опыт на новые, не встречавшиеся ранее примеры предметной области, — ставить диагнозы заболеваний новым, не встречавшимся ранее пациентам.

После того как будет установлена адекватность нейросетевой математической модели и рассматриваемой предметной области, т. е. появится уверенность, что персептрон правильно ставит диагнозы пациентам, в том числе новым, которых в обучающем множестве не было, можно проводить исследования предметной области путем выполнения экспериментов над нейросетевой математической моделью. Например, поставив с помощью персептрона какому-либо пациенту диагноз, можно попробовать поварьировать его входные параметры. Так, увеличив возраст пациента на 5 или 10 лет и введя его в качестве входного параметра персептрона, можно спрогнозировать, как будут развиваться его заболевания в будущем. Можно попытаться улучшить прогнозные диагнозы пациента путем изменения некоторых входных параметров, например смоделировать изменение образа жизни пациента: отказаться от злоупотреблений кофе, курением, наркотиками, изменить диету, заняться спортом, сбросить вес, понизить с помощью лекарственных препаратов артериальное давление и т. д. В результате наблюдения за соответствующими изменениями-откликами выходного вектора можно выполнить прогнозирование состояния здоровья пациента в зависимости от изменяемых входных параметров и разработать на основании этих прогнозов полезные рекомендации: оптимальный образ жизни и оптимальный курс лечения больного.

Один из интересных примеров подобного рода исследований выполнен пермскими учеными [69, 108, 109]. Нейросетевая математическая модель позволила на количественном уровне исследовать известный в медицине факт влияния употребления алкоголя на склонность к заболеваниям сердечно-сосудистой системы. Оказалось, что этот эффект зависит от таких параметров пациента, как рост, индекс массы, возраст, пол, курение, занятия спортом и др., и в каждом конкретном случае он может проявляться не только с отрицательным, но и с положительным результатом.

То же самое выяснилось с соблюдением гипохолестериновой диеты, употреблением кофе, занятиями физической культурой, снижением веса, умственных и физических нагрузок и пр. Оказалось, что заключение о пользе или вреде подобных рекомендаций зависит от сочетания большого комплекса входных параметров системы, характеризующих конкретного пациента и его состояние здоровья.

Надо отметить, что обнаруженные в [108, 109] с помощью нейросетевой диагностико-прогностической системы закономерности не согласуются со сложившейся в современной медицине практикой давать

одни и те же рекомендации всем без исключения кардиологическим больным: соблюдать гипохолестериновую диету, отказаться от «вредных привычек», ограничить употребление кофе, спиртных напитков, похудеть, ограничить умственную и физическую нагрузку и т. д. Компьютерные эксперименты показали, что данная медицинская практика нуждается в пересмотре[1]. Эти рекомендации действительно полезны, но только не для всех, а для большинства больных. Результаты нейросетевого моделирования показали, что указанные рекомендации для некоторых больных, количество которых составляет от 3 до 15%, не только не полезны, но и могут причинить вред. Выявить таких людей позволяют нейросетевые диагностико-прогностичекие системы [108, 109], способные анализировать сотни параметров пациента.

Таким образом, мы можем констатировать факт, имеющий важное методологическое значение: нейросетевая система, обученная на реальном медицинском опыте (без привлечения традиционных академических медицинских знаний), смогла выявить закономерности, неизвестные медицинской науке. Эти новые закономерности можно учитывать в медицинской практике для оптимизации образа жизни и лечения больных.

Обратиться к разработке пермских ученых — интеллектуальной диагностико-прогностической системе, проверить состояние здоровья и выполнить прогнозы развития обнаруженных заболеваний сердечно-сосудистой системы, а затем постараться улучшить эти прогнозы за счет изменения образа и условий жизни можно из раздела «Проекты» сайта Пермского отделения Научного совета РАН по методологии искусственного интеллекта **www.PermAI.ru**.

В заключение еще раз укажем причины, на основании которых можно ожидать, что искусственный нейросетевой врач превзойдет врача-человека.

Во-первых, качество работы искусственного врача всегда стабильно и не зависит от его настроения и состояния здоровья. Во-вторых, и это главное, нейронная сеть способна извлекать и применять знания, которые современной науке неизвестны. Врач обладает только теми знаниями, которые преподаются в медакадемиях и имеются в медицинской литературе. Нейронная же сеть извлекает знания из реального практического материала, который богаче теоретического.

Поэтому есть основания ожидать, что, благодаря применению методов интеллектуального анализа данных, в будущем несовершенство современной медицины будет в значительной степени ликвидировано.

[1] Этот научный результат, полученный методом нейросетевого интеллектуального анализа данных, хотя вступает в некоторое противоречие с традиционными медицинскими представлениями, однако был признан научной общественностью, что подтверждается публикациями в авторитетных рецензируемых научных журналах медицинского профиля [108, 109].

Есть основания ожидать, что применение нейросетевых технологий в медицине позволит по-иному подойти к проблеме получения новых медицинских знаний.

Есть основания ожидать, что в недалеком будущем эффективным инструментом получения новых научных знаний в медицинской науке (и не только в ней) станут диагностико-прогностические системы, построенные на принципах работы нейронных сетей и нейрокомпьютеров.

Контрольные вопросы и задания к § 5.2

1. Когда возник метод математического моделирования?
2. Приведите примеры применения метода математического моделирования.
3. Чем отличается методика построения математических моделей на основе нейротехнологий от традиционной?
4. Опишите, как бы вы стали формировать примеры для обучения персептрона ставить диагнозы заболеваний.
5. Сколько ваш персептрон должен иметь входов и выходов?
6. Как быть, если врач-эксперт не совсем уверен в правильности выставляемого им диагноза?
7. Почему нейросетевой врач может превзойти обычного врача по качеству постановки диагнозов заболеваний?
8. Откуда нейросетевой врач получает медицинские знания, и в каком виде он хранит их в своей памяти?
9. Зайдите на сайт **www.LbAi.ru** и выполните лабораторные работы № 6, 7 — обучите персептрон ставить диагнозы заболеваний: «Пневмония», «Острое респираторное заболевание (ОРЗ)», «Здоров». В случае затруднений или заинтересованности обратитесь к учебно-методическим пособиям [73, 88].
10. Зайдите на сайт **www.PermAI.ru** в раздел «Проекты» и с помощью интеллектуальной прогностико-диагностической системы попробуйте проверить состояние сердечно-сосудистой системы ваше, вашего отца, мамы, дедушки, бабушки, знакомых. Выполните прогноз его/ее состояния здоровья на ближайшие 2, 3, 4, 5 лет. Выясните, как следует изменить образ вашей жизни (сбросить или набрать вес, заняться спортом или лечебной физкультурой, отказаться от курения и алкоголя, следить за давлением и т. д.), а также жизни ваших родственников и знакомых, чтобы улучшить этот прогноз.

5.3. ДИАГНОСТИКА ТЕХНИЧЕСКИХ УСТРОЙСТВ

Не касаясь разнообразных применений нейронных сетей для управления промышленными роботами и технологическими процессами, рассмотрим важную техническую проблему диагностики неисправно-

стей сложных технических устройств. Считается, что эта проблема проще рассмотренной выше проблемы диагностики здоровья человека. Действительно, если строить диагностическую систему методами традиционного математического моделирования, основанного на законах природы — физики, химии, биологии и др., то это утверждение справедливо. Дело в том, что для технического устройства, каким бы сложным оно ни было, всегда известно функциональное назначение любого сколь угодно мелкого узла, что нельзя сказать о человеке. Естественно, что это обстоятельство является серьезным препятствием, усложняющим построение адекватных математических моделей человека традиционными способами. Нейронная же сеть сама извлекает необходимые знания из обучающих примеров, сама устанавливает неизвестные законы и раскрывает функциональные связи между элементами исследуемого объекта. Поэтому в случае нейросетевого моделирования нет принципиальной разницы между диагностикой заболеваний человека и диагностикой неисправностей технического устройства.

Нейросетевой подход к решению задач медицинской диагностики, рассмотренный ранее, можно почти без изменений применить к решению проблемы диагностики неисправностей сложных технических устройств. Продемонстрируем это на примере диагностики неисправностей авиационных двигателей.

Специалисты, занимающиеся этой проблемой, устанавливают датчики, измеряющие параметры работы авиадвигателей во время полетов. Файл данных полетного мониторинга обычно содержит такие параметры, как номер полета, дата полета, общая наработка двигателя, температура и давление воздуха на входе в двигатель, температура и давление газа за турбиной, температура лопаток, уровень и температуру масла в маслоблоке и т. д. Число полетных параметров может достигать сотни и более, что, кстати, соизмеримо с числом параметров больного при постановке медицинского диагноза его заболевания.

После выполнения определенного количества полетов (обычно порядка тысячи) двигатель снимают с самолета и подвергают стендовой разборке (рис. 5.2), во время которой выявляют и устраняют его дефекты. Характерными дефектами авиадвигателей являются: трещина на сопловом агрегате, забоины, разрушение дефлектора, разрушение подшипников, наличие стружки в масле (всего около 30 видов дефектов).

Задача инженера-диагноста состоит в том, чтобы, используя данные мониторинга, выявить дефекты двигателя до его профилактической разборки. Традиционно эта задача решается с помощью методик, основанных на физических закономерностях: каждый дефект вызывает определенные отклонения тех или иных полетных параметров работы двигателя, поэтому, анализируя характер их изменения, можно сделать предположения о появлении дефектов, вызывающих эти изменения. Понятно, что ввиду значительных объемов информации и сложно-





Рис. 5.2. После выполнения определенного количества полетов двигатель снимается с самолета и подвергается стендовой разборке

сти существующих взаимосвязей между дефектами и измеренными параметрами задача анализа данных полетного мониторинга и выявления дефектов авиадвигателей является далеко не тривиальной и, бывает, решается ненадежно и некачественно. В результате это приводит либо к неоправданно ранней отправке двигателя в ремонт, либо к трагическим последствиям.

Рассмотрим, как можно решать эту задачу с помощью нейросетевого моделирования. Прежде всего отметим, что во входном векторе персептрона $\boldsymbol{X}$ следует предусмотреть места для всех параметров полетного мониторинга, на значения которых оказывают влияние выявляемые дефекты. Возможные дефекты авиадвигателя можно закодировать в выходном векторе $\boldsymbol{D}$ с помощью все тех же нулей и единиц. Векторы желаемых выходов $\boldsymbol{D_q}$ составляются по результатам стендовых разборок двигателей. В отличие от медицинской диагностики здесь нет необходимости вводить многобалльную шкалу диагнозов, поскольку все диагнозы в обучающем множестве примеров имеют 100%-ю достоверность, т. е. ситуация аналогична случаю, когда диагноз болезни ставит патологоанатом при вскрытии тела пациента в морге.

Кроме того, разумно предположить, что при первых вылетах нового или отремонтированного самолета его двигатель полностью исправен и дефектов нет, тогда как при последних вылетах двигатель уже имел те самые дефекты, которые выявились при его разборке. Поэтому из всего множества параметров полетного мониторинга ценность для обучения нейронной сети имеют параметры первого и последнего полетов самолета. Таким образом, для каждого двигателя, побывавшего на разборке, формируется пара обучающих векторов $\boldsymbol{X_q}$ и $\boldsymbol{D_q}$.

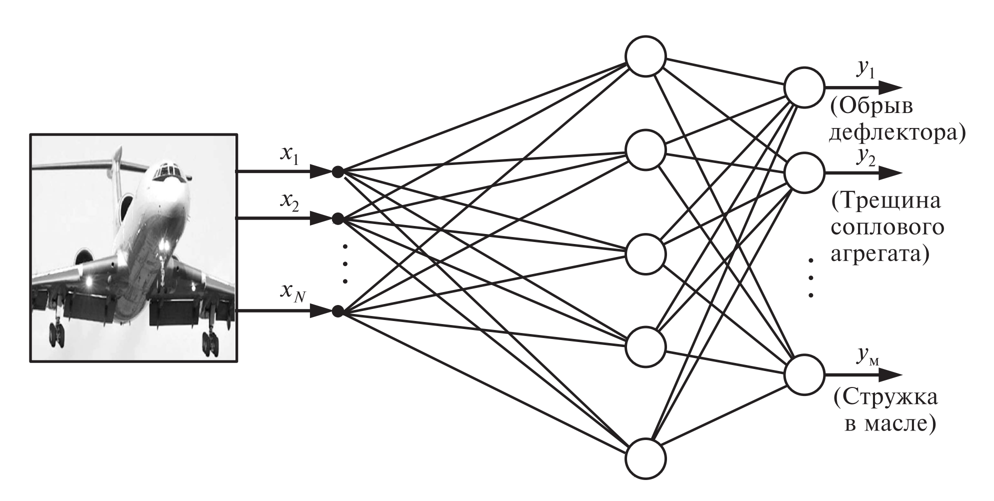


Рис. 5.3. Принципиальная схема нейросетевой системы диагностики авиационного двигателя

Если количество обучающих примеров будет достаточным, то правильно спроектированный многослойный персептрон (рис. 5.3) обучится надежно ставить диагнозы неисправностей авиационных двигателей, в том числе и тех, которые в обучающем множестве не участвовали. А если таким персептроном оборудовать бортовой компьютер самолета, то он будет сообщать о появлении дефекта двигателя в реальном времени, т. е. как только сложится соответствующая конфигурация вектора входных параметров — результатов измерений, снимаемых во время полета самолета. Естественно, что прежде, чем попасть в кабину пилотов, сигналы персептрона должны обрабатываться компьютером, вырабатывающим инструкции об адекватных действиях экипажа при создавшихся условиях.

Укажем на еще одно преимущество нейросетевого метода диагностики перед традиционными диагностическими методами, основанными на явных знаниях. Лучше всего это сделать на примере случая, однажды происшедшего при выполнении проекта создания системы диагностики авиационных двигателей [30].

Прежде всего заметим, что специалисты, занимающиеся традиционными методами диагностики технических устройств, обычно пытаются учесть как можно больше взаимосвязей между значениями измеряемых полетных параметров и появлением тех или иных дефектов двигателя. В сложных технических устройствах количество таких взаимосвязей настолько велико, что построить полную математическую модель, абсолютно адекватную моделируемому устройству, традиционными методами математического моделирования практически невозможно. Кроме того, в сложных технических устройствах существуют

и такие взаимосвязи, о которых специалисты не знают. Так, в двигателестроении известен дефект, называемый «стружка в масле» — при работе трущихся частей иногда образуется мелкая стружка, которая падает в поддон с маслом. Специалистам известно, что возникновение этого дефекта не влияет ни на один из измеряемых датчиками параметров работы двигателя, вследствие чего не существует традиционных методик, способных выявлять этот дефект. Следовательно, и нейросеть, на вход которой подаются именно эти параметры двигателя, принципиально не должна ставить диагноз «стружка в масле». Но, как зафиксировано в работе [30], нейронная сеть этот диагноз ставила, и оказывалась права!

Складывалось впечатление, что нейронная сеть принимает правильные решения, используя информацию, недостаточную с точки зрения традиционной логики. В жизни это свойство мозга принято называть «интуицией» или «шестым чувством». В связи с этим в работе [63] было высказано мнение, что нейронная сеть, выполненная «по образу и подобию» мозга, унаследовала от него и это замечательное свойство человека Но после обсуждений и дискуссий со специалистами парадокс объяснили тем, что, по-видимому, не оказывая заметного влияния на каждый отдельно взятый параметр двигателя, появление стружки все-таки влияет на общую конфигурацию входного вектора, что и вызывает соответствующую реакцию нейронной сети.

Впоследствии мы не раз наблюдали, как нейросети ставили правильные диагнозы и выдавали подтвердившиеся затем прогнозы, объяснить которые в рамках традиционной логики было затруднительно, а термины «интуиция» и «шестое чувство нейронных сетей», введенные и продемонстрированные в книге [63], стали употребляться многими авторами.

Контрольные вопросы и задания к § 5.3

1. Опишите, как бы вы стали формировать примеры для обучения персептрона ставить диагнозы неисправностей автомобильного двигателя.
2. Сколько ваш персептрон должен иметь входов и выходов?
3. Может ли нейросетевая диагностическая система превзойти традиционную диагностическую систему, основанную на явных знаниях (экспертную систему) по качеству постановки диагнозов? Объясните почему.
4. Откуда нейросетевая диагностическая система получает технические знания, и в каком виде она их хранит в памяти?
5. Что понимается под термином «Интуиция нейронных сетей»?

5.4. ДИАГНОСТИКА ЭКОНОМИЧЕСКОГО СОСТОЯНИЯ ПРЕДПРИЯТИЙ

Сегодня в мире существует множество различных методик прогнозирования банкротств предприятий. В то же время, по мнению специалистов, единого общепринятого метода диагностики экономического состояния предприятий в настоящее время не существует. Кроме того, как отмечают многие российские авторы, многочисленные попытки применения иностранных моделей прогнозирования банкротств в отечественных условиях не позволили получить достаточно точные результаты.

Авторами работы [70] была предпринята попытка создания нейросетевой системы, оценивающей вероятность банкротства российских банков, основанной на закономерностях российской действительности, отраженных в финансовой отчетности ЦБ РФ. В качестве входных параметров нейросетевой математической модели были выбраны: достаточность собственных средств; коэффициент мгновенной ликвидности; коэффициент текущей ликвидности; коэффициент долгосрочной ликвидности; рентабельность активов; город, в котором расположен банк; правовая форма; размер уставного капитала; размер активов; показатель использования собственных средств для приобретения акций других юридических лиц и др., всего 15 параметров. Выходная переменная y принимает значение, равное «единице», если у банка была отозвана лицензия, и значение «нуль», если банк продолжает свою деятельность. Было сформировано множество, содержащее 111 примеров, из которых 52 примера являются данными обанкротившихся банков, 59 примеров относятся к работающим банкам. Все множество примеров разбито на обучающее, использованное для обучения сети, и тестирующее, предназначенное для проверки ее прогностических свойств. Естественно, что примеры тестирующего множества при обучении сети не использовались.

Оптимальная структура нейронной сети представляла собой персептрон, имеющий 15 входных нейронов, один скрытый слой с пятью нейронами и один выходной нейрон. В качестве активационных функций нейронов скрытого слоя и выходного нейрона использовались сигмоидные функции.

После обучения прогностические свойства сети проверялись на тестирующих примерах. Изначально среднеквадратичная ошибка тестирования нейронной сети составила 13,5%, поэтому из множества примеров вручную были удалены банки-банкроты, причиной отзыва лицензий которых была некорректно составленная финансовая отчетность. Дело в том, что эти причины не являются объективными, а потому данные о таких банках выпадают из общих статистических закономерностей. После удаления подобных данных среднеквадратичная ошибка тестирования составила 6,3%.

Затем в процессе обучения и тестирования нейросети были обнаружены выбросы, выявленные с помощью методики (см. § 6.5, а также [57]). В число выбросов, обнаруженных с помощью этой методики, попали данные о двух банках: «Сембанк» и «Анкор-банк». Данные о первом банке были квалифицированы нейронной сетью как выброс, по-видимому, в связи с тем, что у него была отозвана лицензия, хотя банк имел хорошие показатели достаточности капитала и коэффициенты ликвидности. Второй банк, наоборот, не подвергся санкциям ЦБ РФ, хотя имел довольно низкие показатели достаточности капитала и самый маленький размер уставного капитала из всей выборки.

После удаления обнаруженных нейросетью выбросов и обучения сети среднеквадратичная ошибка тестирования (обобщения) снизилась с 6,3% до 2,6%.

Исследование нейросетевой математической модели выполнялось путем проведения виртуальных компьютерных экспериментов. В ходе этих экспериментов вычисления с помощью нейронной сети выполнялись при изменении одного из входных параметров, а остальные параметры «замораживались», т. е. сохранялись неизменными. В процессе исследования нейросетевой модели решались задачи выявления закономерностей рассматриваемой предметной области — банковской системы России. Мы пользовались тем, что обученная нейросетевая модель реагирует на изменение входных параметров и ведет себя так же, как вела бы себя сама предметная область.

Рассматривались характеристики двух банков, первый из которых изначально классифицировался как банкрот, а второй банк — как не банкрот. Банки существенно отличаются показателем использования собственных средств для приобретения акций других юридических лиц, рентабельностью активов и размером уставного капитала. Как видно из рис. 5.4, на котором приведены результаты вычислительных экспериментов, на вероятность признания банка банкротом существенно влияет коэффициент долгосрочной ликвидности, причем кривые имеют U-образную форму. Повышение коэффициента долгосрочной ликвидности до определенного уровня оказывает положительное влияние на деятельность банка, вероятность банкротства уменьшается. Однако с определенного момента при дальнейшем повышении данного показателя вероятность банкротства увеличивается. Оптимальным показателем долгосрочной ликвидности для изначально надежного рассматриваемого банка оказалось значение 40, а для банка банкрота — 50.

Дальнейшие вычислительные эксперименты показали, что на экономическое состояние банков заметное влияние оказывают и другие параметры, например их организационно-правовая форма, место расположения и др. Однако влияние отдельно взятых параметров неоднозначно и в каждом конкретном случае может проявляться по-разному, в зависимости от множества других параметров, характеризующих банк.

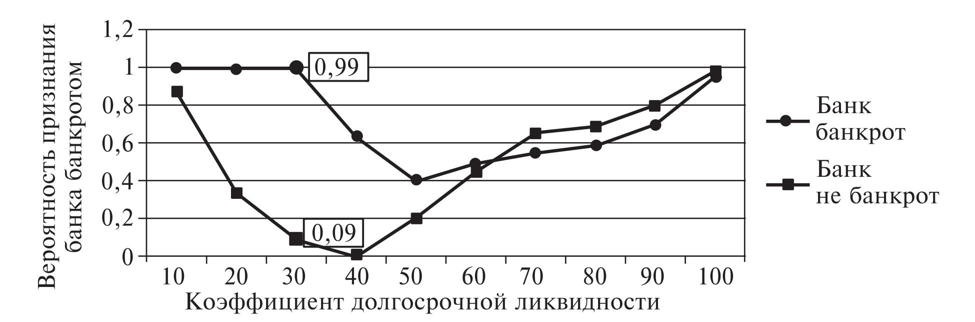


Рис. 5.4. Влияние коэффициента долгосрочной ликвидности на вероятность признания банка банкротом. Маркером увеличенного размера отмечено реальное состояние исследуемых банков

Интеллектуальная система прогнозирования вероятности банкротства российских банков может быть использована для оценки рисков межбанковских кредитов как методика проведения внутреннего аудита, а также для поддержки принятия решений, направленных на совершенствование деятельности банков. Воспользоваться системой можно из раздела «Проекты» сайта **www.PermAI.ru**.

5.5. УПРАВЛЕНИЕ КИБЕРНЕТИЧЕСКИМ ОБЪЕКТОМ

Нейросеть, используемую для управления поведением робота, искусственного спутника Земли, беспилотного самолета, какого-либо кибернетического объекта или персонажа компьютерной игры, называют ***нейроконтроллером***. Управляемый объект находится в окружающей среде, которая постоянно изменяется. Заранее предусмотреть все возможные ситуации невозможно. Идея применения нейросети состоит в том, чтобы обучить ее на конечном числе ситуаций, а затем предоставить нейросети возможность самостоятельно реагировать на ситуации, которых в обучающем множестве примеров не было.

Допустим, речь идет о компьютерной игре, в которой персонаж, в зависимости от ситуации, может предпринять одно из четырех действий:

1) атаковать;
2) передвигаться;
3) прятаться;
4) убегать.

Выбор действия зависит от того, сколько врагов видит персонаж, насколько хорошо он вооружен и какое у него здоровье в данный момент. В табл. 5.1 приведено множество обучающих примеров: в первом столбце (x_1) помещена информация о количестве врагов,

Таблица 5.1

Множество обучающих примеров действий персонажа компьютерной игры

Кол-во врагов (x_1)	Персонаж имеет нож (x_2)	Персонаж имеет пистолет (x_3)	Здоровье персонажа (x_4)	Действие персонажа (d)
0	0	1	1	2 (передвигаться)
1	0	1	1	1 (атаковать)
1	0	0	1	4 (убегать)
1	0	0	0	3 (прятаться)
1	1	1	0	3 (прятаться)
1	1	0	1	1 (атаковать)
0	1	1	0	2 (передвигаться)
5	1	1	2	1 (атаковать)
10	1	1	2	4 (убегать)

во втором и третьем (x_2 и x_3) — информация о вооружении персонажа, в четвертом (x_4) — информация о его здоровье (0 — плохое, 1 — удовлетворительное, 2 — хорошее) и в пятом (d) — действие, которое следует предпринять персонажу.

Персептрон, снабженный четырьмя входными и одним выходным нейронами и обученный на примерах табл. 5.1, способен генерировать вполне адекватные рекомендации о действиях персонажа, причем он «не теряется» и в таких непредвиденных ситуациях, которых во множестве обучающих примеров не было.

Еще раз обратим внимание, что в качестве управляемого может быть любой кибернетический объект, например искусственный спутник Земли, баллистическая ракета, самолет-беспилотник, глубоководный аппарат, который управляется автономно или с которым периодически теряется связь и который вследствие этого сам должен принимать решения о своем дальнейшем поведении

5.6. ПРОГНОЗИРОВАНИЕ ПОЛИТИЧЕСКИХ СОБЫТИЙ

Вероятность наступления того или иного политического события зависит от многих факторов. Например, результаты выборов президента страны зависят от экономической и политической обстановки в мире, активности средств массовой информации, активности и степени политизированности самих избирателей, взаимоотношений и политических ходов партий, качества предвыборной компании, фальсификаций и многого другого. Учесть и правильно оценить влияние всех этих факторов на исход выборов чрезвычайно сложно. Но есть и такие

факторы, которые определяются только самой личностью претендента в президенты. Это его возраст, пол, семейное положение, количество детей, род занятий, социальная принадлежность и др. Можно ли при построении прогностической программы ограничиться учетом только этих личностных характеристик претендента?

Одна из интереснейших попыток интеллектуального анализа данных с целью ответа на этот вопрос была предпринята студентами пермских университетов и описана в работах [71, 82, 89]. В экспериментах использовалась нейронная сеть с пятью входами: x_1 — возраст претендента в президенты; x_2 — его происхождение (1 — из бедных, 2 — средний класс, 3 — интеллигенция, 4 — из богатых); x_3 — семейное положение (1 — женат, 0 — холост); x_4 — количество детей; x_5 — основная область деятельности (1 — военная, 2 — политическая, 3 — общественная, 4 — экономическая). Сеть имела один выходной нейрон, выходной сигнал которого d мог принимать значение 1, означающий победу претендента на выборах и 0 — поражение.

Множество примеров, взятых из истории президентских выборов США и Франции, было разбито на обучающее (табл. 5.2), использованное для обучения сети, и тестирующее (табл. 5.3), предназначенное для проверки ее прогностических свойств. Естественно, что примеры тестирующего множества при обучении сети не использовались.

В таблице 5.3 в предпоследнем столбце d помещены желаемые выходы нейросети, т. е. те, которые имели место в истории США и Франции, а в последнем столбце — прогноз сети y, обученной на множестве примеров табл. 5.2. Как видно из табл. 5.3, нейросеть, хотя и с погрешностью, совершенно правильно «предсказала» победу Дж. Кеннеди, А. Линкольну и Д. Эйзенхауэру и поражение — М. Дукакису и Дж. Макговерну. Это значит, что она смогла извлечь из весьма скромного обучающего множества некую скрытую закономерность и применить ее при тестовом прогнозировании.

Невероятно, но факт! Получается, что рейтинг претендентов в президенты стран практически не зависит от политической конъюнктуры, лозунгов, обещаний, политических платформ. Получается, что результат голосований в первую очередь определяется неким удачным сочетанием его субъективных личностных характеристик, которое выявила нейронная сеть.

Студентами пермских университетов была введена еще одна малообоснованная гипотеза. Они приняли допущение, что менталитет россиян не отличается от менталитета французов и американцев и на предстоящих через полтора года выборах президента своей страны россияне будут отдавать голоса, руководствуясь аналогичными критериями.

Собрав необходимые личностные данные на возможных российских претендентов (табл. 5.4) и применив к ним обученную на французском и американском опытах нейросеть, пермские студенты выполнили

Таблица 5.2

Обучающее множество примеров президентских выборов в США и Франции

Претенденты	Возраст	Происхож-дение	Семейное положение	Кол-во детей	Деятель-ность	Победа или поражение
	x_1	x_2	x_3	x_4	x_5	d
А. Гор	52	3	1	3	2	0
Б. Голдуотер	58	4	1	2	2	0
Б. Клинтон	46	2	1	2	2	1
В. Жискар	55	2	1	2	2	0
Д. Вашингтон	57	1	1	0	1	1
Д. Адамс	62	3	1	1	2	1
Д. Буш-мл.	54	4	1	2	4	1
Д. Буш-ст.	64	4	1	1	4	1
Д. Бьюкенен	66	4	0	0	2	1
Д. Доул	73	2	0	0	1	0
Д. К. Адамс	58	3	1	2	2	1
Д. Керри	61	4	1	2	2	0
Д. Медисон	58	4	1	0	2	1
Д. Монро	49	1	1	2	1	1
Д. Тейлер	51	3	1	15	2	1
Д. Форд	61	2	1	2	1	1
Ж. М. Ле Пен	74	1	1	3	3	0
Ж. Ширак	63	2	1	2	2	1
Л. Жоспен	58	2	1	0	2	0
Р. Рейган	70	2	1	2	2	1
Т. Джеферсон	58	4	1	1	2	1
У. Мондейл	56	2	1	3	2	0
Ф. Миттеран	75	2	0	1	3	1
Ф. Пирс	49	4	1	0	2	1
Ш. де Голль	68	3	1	3	1	1

прогноз президентских выборов в России на 2008 г. (см. последний столбец табл. 5.4).

На рисунке 5.5 эти прогнозы представлены в виде гистограммы. Если полученные в результате прогноза цифры рассматривать как сравнительный рейтинг претендентов, то можно заключить, что наибольшие шансы стать президентом России нейронная сеть отдала Д. А. Медведеву, а наименьшие — Ю. П. Трутневу. Рейтинги осталь-

Таблица 5.3

Тестирующее множество примеров президентских выборов

Претенденты	**Возраст**	**Происхож-дение**	**Семейное положение**	**Кол-во детей**	**Деятель-ность**	**Победа или поражение**	**Прогноз сети**
	x_1	x_2	x_3	x_4	x_5	d	y
Дж. Кеннеди	44	4	1	2	2	1	0,82
А. Линкольн	52	1	1	4	1	1	0,93
М. Дукакис	56	2	1	2	2	0	0,00
Дж. Макговерн	59	2	1	0	2	0	0,11
Д. Эйзенхауэр	63	2	1	2	1	1	1,04

Таблица 5.4

Личностные характеристики возможных претендентов на пост президента России и прогноз результатов выборов на 2008 г.

Возможные претенденты	**Возраст**	**Происхож-дение**	**Семейное положение**	**Кол-во детей**	**Деятель-ность**	**Прогноз сети**
	x_1	x_2	x_3	x_4	x_5	y
С. Б. Иванов	55	2	1	2	1	0,91
Д. А. Медведев	42	2	1	1	3	1,59
Б. В. Грызлов	58	2	1	2	4	1,01
Г. А. Зюганов	64	1	1	2	2	1,04
В. В. Жириновский	62	1	1	3	2	0,79
Д. О. Рогозин	45	3	1	1	2	1,14
Ю. П. Трутнев	52	2	1	2	2	0,13

ных претендентов, занимая промежуточное положение, различаются между собой незначительно.

Еще раз обратим внимание, что данный прогноз выполнен и опубликован в книге [71] в начале 2007 г., т. е. более чем за год до выборов Президента России, состоявшихся в 2008 г. Политическая обстановка в то время складывалась так, что о преподавателе вуза, политическом и общественном деятеле Дмитрии Медведеве вообще мало кто слышал. Поэтому данные нейросетевого прогнозирования хотя и были опубликованы, однако с комментариями, что к этому прогнозу «не стоит относиться серьезно», так как «прогноз выполнен в предположении, что избирательские менталитеты французов, американцев и россиян совпадают, что, по-видимому, неверно. Такой рейтинг был бы у претендентов на пост президента не в России, а в Америке или во Франции».

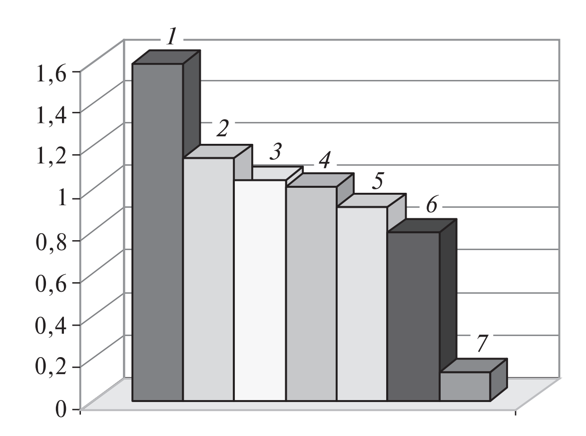


Рис. 5.5. Прогноз президентских выборов в России на 2008 г., выполненный студентами пермских университетов в начале 2007 г. [71]

1 — Д. А. Медведев; *2* — Д. О. Рогозин; *3* — Г. А. Зюганов; *4* — Б. В. Грызлов; *5* — С. Б. Иванов; *6* — В. В. Жириновский; *7* — Ю. П. Трутнев

Впоследствии Д. А. Медведев действительно победил на выборах 2008 г. и стал президентом России. Как видим, мы напрасно сомневались в результатах нейросетевого прогнозирования. Нейронная сеть оказалась права. Прогноз результатов президентских выборов, выполненный студентами пермских университетов, подтвердился, что можно считать серьезным успехом метода интеллектуального анализа данных на основе нейросетевого математического моделирования.

Методы нейросетевого моделирования позволяют не только предсказывать будущие события, но и исследовать их, ставить над ними виртуальные эксперименты. Изменяя входные параметрами обученной нейронной сети и производя вычисления, можно получить ответы на многие вопросы, интересующие политтехнологов. Например, изменяя возраст политического деятеля и производя вычисления с помощью нейросети, можно проследить, как будет меняться его рейтинг со временем, обнаружить пики его популярности среди избирателей и выбрать момент, когда ему лучше всего уйти в отставку. Так, в работах [82] и [89], опубликованных в 2008 и 2010 гг., когда президент Д. А. Медведев был на вершине своей популярности, нейронные сети прогнозировали ему снижение рейтинга (рис. 5.6, *а*), тогда как другому политику — В. В. Жириновскому — прогнозные кривые предсказали постепенный рост популярности (см. рис. 5.6, *б*), что и наблюдалось в последующие годы.

Проводя целенаправленные исследования нейросетевой модели, можно разработать рекомендации, каким образом тому или иному политическому деятелю изменить свой рейтинг. На рисунке 5.7 приведены результаты моделирования различных возможных изменений

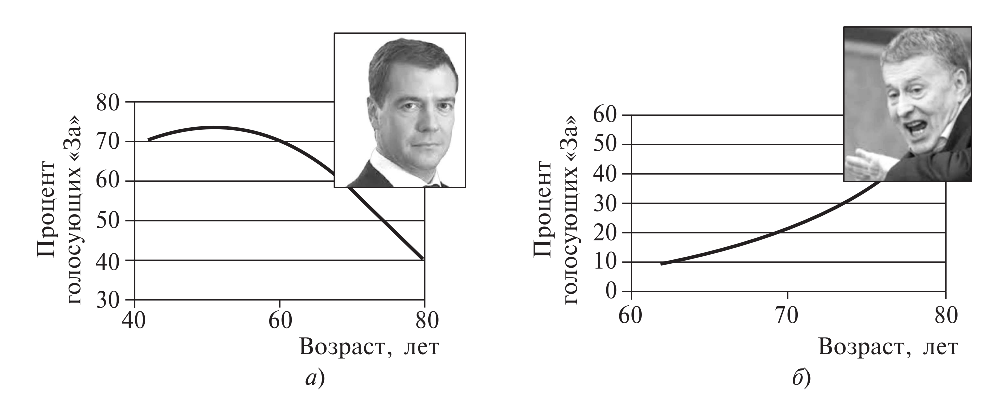


Рис. 5.6. Зависимость рейтинга Д. А. Медведева (*а*) и В. В. Жириновского (*б*) от их возраста. Прогноз опубликован в 2008 г [82], когда президент Д. А. Медведев был на вершине своей популярности

в личной и политической жизни известного политического деятеля, кандидата в президенты России на выборах 2004 г. Ирины Хакамады.

Как видно из гистограммы рис. 5.7, повторное участие в выборах увеличит число голосующих «За» до 8,4% (*2*) (против исходных 3,5% (*1*)), а вот третья попытка стать президентом страны, наоборот, приведет к снижению количества голосов до 1,7% (*3*). Смена партии на «правящую» приведет к падению ее шансов до нуля (*4*). Если Ирина Хакамада сменит свою деятельность с экономической на спорт или культуру, то ее рейтинг увеличится до 8,5% голосов (*5*). Если она изберет военную карьеру, то это принесет ей 17,5% голосов (*6*), а если станет преемником президента страны, то за нее будут голосовать 67,4% избирателей (*6*). Если она разведется, то ее шансы победить на выборах упадут до нуля (*8*). Если сменит пол, то ее рейтинг повысится до 5,5% голосов (*9*). Если заведет еще одного ребенка, то рейтинг возрастет до 9,6% голосов (*10*). Если она заведет двоих детей, то рейтинг повысится до 16,5% (*11*), а если троих, то до 24,1% (*12*). Если И. М. Хакамаде удастся радикальным образом повысить свою популярность, то она имеет шансы получить 47,8% голосов российских избирателей (*13*).

В заключение отметим, что, помимо прогнозирования президентских выборов, нейросетевые математические модели могут быть использованы для прогнозирования исходов любых видов голосований, например выборов ректора вуза, выборов в законодательные собрания областей и краев, в академики РАН и др. Они могут быть использованы для выявления перспективных кандидатов, а также для разработки рекомендаций по изменению их рейтингов.

Итак, применение метода интеллектуального анализа данных на основе нейросетевого моделирования позволяет выявлять и анализировать закономерности такого сложного политического явления, как

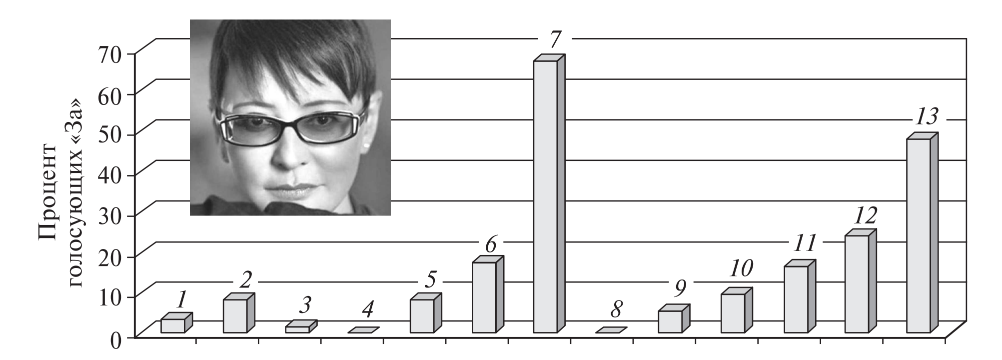


Рис. 5.7. Результаты моделирования влияния возможных изменений в личной и политической жизни И. М. Хакамады на ее рейтинг

1 — исходное состояние; *2* — второй раз участвовать в выборах; *3* — третий раз участвовать в выборах; *4* — сменить партию на «правящую»; *5* — сменить деятельность на спорт или культуру; *6* — сменить деятельность на военную; *7* — стать преемником президента; *8* — развестись; *9* — сменить пол на мужской; *10* — завести еще одного ребенка; *11* — завести еще двоих детей; *12* — завести еще троих детей; *13* — радикально повысить свою популярность

президентские выборы. Причем нейросетевые математические модели предназначены не только для пассивного предсказания будущих событий — побед или поражений кандидатов. Исследуя эти модели и выявляя закономерности, можно научиться понимать, каким образом следует оказывать влияние на эти будущие события, разворачивать их траекторию развития в нужном направлении, создавать благоприятную ситуацию для достижения своих целей.

Контрольные вопросы и задания к § 5.6

1. Какая информация использовалась для создания обучающего множества примеров при построении программы, прогнозирующей результаты президентских выборов в России?
2. Для чего нужно тестирующее множество примеров и чем оно отличается от обучающего множества примеров?
3. Попробуйте создать программу, прогнозирующую результаты президентских выборов в России, используя в качестве обучающих примеров отечественный опыт. Постройте зависимость рейтинга президента от его возраста. Попробуйте сместить пик популярности президента на более зрелый возраст за счет изменения его параметров.
4. Каким образом можно использовать нейросеть для разработки рекомендаций политическим деятелям по повышению их рейтинга?

5.7. ВЫЯВЛЕНИЕ СПОСОБНОСТИ ЧЕЛОВЕКА К БИЗНЕСУ

В педагогической науке нейросети могут найти применение, например, при решении таких вопросов, как: какую профессию выбрать? куда пойти учиться?

Начнем с выбора профессии. Кто из нас не решал эту проблему? От правильного выбора профессии зависит вся дальнейшая судьба человека.

При выборе профессии очень важно понять и оценить возможности своего организма, свои способности, найти именно ту нишу в жизни, ту область деятельности, где могут максимальным образом раскрыться способности человека, заложенные в него природой.

В наше время многие молодые люди желают в будущем стать бизнесменами. Но ученые утверждают, что к предпринимательской деятельности в среднем способны не более 7–10% населения. И что ждет тех, кто выберет этот путь, не зная, что природа не отнесла его к этой золотой десятке процентов: банкротства? разорения? разочарования?

История знает немало примеров [68], когда маститые ученые, авторы гениальных открытий, пытались внедрять свои открытия, организовать на их базе инновационный бизнес, но терпели неудачу, в то время как другие люди, не имеющие столь глубокого образования и интеллекта, с легкостью реализовывали проекты ученых.

Значительное число исследователей придерживается мнения, что научить предпринимательству нельзя, что для этого рода деятельности необходимы особые способности. Природной же предпосылкой развития способностей является наличие *задатков* — некоторых врожденных анатомо-физиологических особенностей мозга и нервной системы. К задаткам, согласно позиции известного отечественного ученого-психолога профессора В. А. Крутецкого, относятся типологические свойства нервной системы, природные свойства анализаторов, индивидуальные варианты функционирования коры головного мозга.

Утверждение, что «предпринимателем надо родиться», подтверждает следующий факт: по обследованиям, проводимым в США и других странах, в 50–72% случаев у основателей фирм по крайней мере один из родителей имеет свободную профессию. В Гарвардскую школу бизнеса, например, принимают людей, обладающих предпринимательским типом личности по специальному тесту Мейерс-Бриггс.

Существует множество попыток научиться выявлять людей, способных к предпринимательству. Так, группа американских ученых провела исследование и доказала, что люди с голубыми глазами намного способнее темноглазых и у них гораздо более развито стратегическое мышление. По мнению многих специалистов в области методов оценки и подбора персонала, существуют зависимости между цветом глаз и успешностью в бизнесе.

Как мы уже неоднократно убеждались, для выявления зависимостей, не поддающихся объяснению, и для построения математических моделей, использующих такие зависимости, перспективно применение методов искусственного интеллекта, в частности аппарата нейросетевых технологий. По-видимому, впервые такая попытка была предпринята авторами работы [84].

Как мы уже знаем, при формировании нейросетевой математической модели большое значение имеет выбор значимых входных параметров, т. е. тех параметров предметной области, которые оказывают влияние на результат моделирования. Можно назвать множество факторов, влияющих на успешность бизнеса. Однако многие из этих факторов не поддаются количественной оценке. Другие факторы невозможно включить в математическую модель ввиду отсутствия информации о значениях требуемых показателей. Авторы работы [84] включили в модель в качестве входных параметров факторы, влияющие на успешность предпринимательской деятельности в разрезе понятия «предпринимательский дух», в основе которого, по мнению множества зарубежных ученых и успешных предпринимателей, лежит понятие «предпринимательской ДНК». В связи с этим основная часть выбранных факторов являлась характеристикой биометрических показателей человека. Принимая во внимания указанные ограничения, в качестве входных параметров нейросетевой математической модели были использованы:

x_1 — возраст предпринимателя;
x_2 — пол;
x_3 — количество детей;
x_4 — образование;
x_5 — сфера деятельности, в которой предприниматель планирует вести бизнес;
x_6 — цвет глаз предпринимателя.

В качестве выходного параметра нейронной сети была принята величина, которую авторы работы [84] назвали вероятностью успешности бизнеса: $d = 1$, если бизнес успешен, и $d = 0$, если неуспешен.

В качестве обучающих примеров была использована информация о бизнесменах из журнала Forbes (выпуск — май 2009 г. «100 богатейших бизнесменов»), данные из сети Интернет и данные социологического опроса. В обучающие множества входили примеры как успешного бизнеса ($d = 1$), так и неуспешного ($d = 0$). После обучения сеть тестировалась на примерах, взятых из тех же источников, но не участвовавших в процессе обучения сети. Максимальная погрешность прогноза на тестовых примерах составила 13,1%. Анализ значимости входных параметров показал, что все они являются важными, поскольку удаление любого из них приводит к резкому увеличению погрешности прогнозов на тестовых примерах.

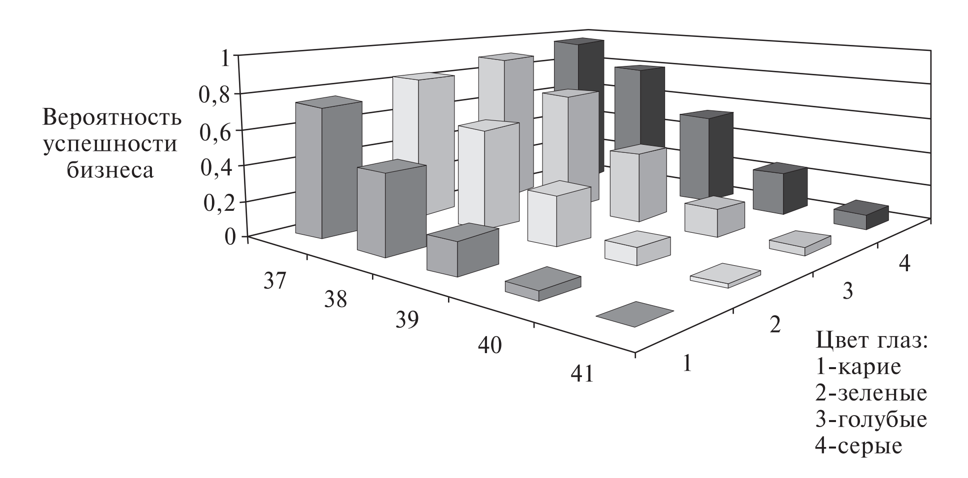


Рис. 5.8. Зависимость вероятности успешности бизнеса от возраста и цвета глаз предпринимателя, являющегося бездетным мужчиной, имеющего высшее образование и желающего заняться бизнесом в сфере финансов

Таким образом, эксперимент [84] можно считать удачным: нейронная сеть хотя и с погрешностью, но усвоила закономерности предметной области и ее можно применять в качестве математической модели этой области. Она подходит как для прогнозирования успешности бизнеса желающих заняться предпринимательской деятельностью, так и для изучения закономерностей этого сложного явления. Производя вычисления с помощью нейронной сети, изменяя при этом одни входные параметры и сохраняя неизменными другие, можно построить зависимости вероятности успешности бизнеса от входных параметров. Одна из таких зависимостей графически представлена на рис. 5.8.

Как видим, наибольшей склонностью к бизнесу в сфере финансов среди бездетных мужчин с высшим образованием имеют обладатели серых глаз, затем идут зеленоглазые, голубоглазые и на последнем месте — обладатели карих глаз. Причем независимо от цвета глаз вероятность успешности бизнеса снижается с возрастом и к 40 годам приближается к нулю.

В результате проведения дальнейших вычислений и построения аналогичных гистограмм были выявлены другие зависимости между входными параметрами и вероятностью успешности предпринимательской деятельности. Было установлено, что цвет глаз может как оказывать влияние, так и совсем не влиять на вероятность успешности предпринимательской деятельности. Так, у выпускников вузов (мужчин), планирующих открыть свое дело в сфере розничной торговли товарами народного потребления, цвет глаз незначительно влияет

на степень успешности предпринимательской деятельности, однако вероятность успешности выше у обладателей карих глаз, затем идут обладатели зеленых и голубых глаз. Самая низкая вероятность успешной организации собственного дела у обладателей серых глаз. Что касается бездетных мужчин с высшим образованием, планирующих организовать свое дело в медицине, то самая высокая вероятность успешности у обладателей серых глаз, затем идут обладатели голубых, зеленых и, наконец, карих глаз.

В зависимости от различных сочетаний входных параметров, были выявлены случаи, когда цвет глаз не оказывал влияния на вероятность успешности предпринимательской деятельности. Так, бездетным мужчинам с высшим образованием, независимо от цвета глаз, не следует заниматься автомобильным бизнесом. Мужчинам с высшим образованием в возрасте до 36 лет, вне зависимости от цвета их глаз, можно рекомендовать открывать бизнес в сфере розничной торговли.

В заключение отметим, что указанные здесь закономерности и следующие из них рекомендации не охватывают всего многообразия знаний, извлеченных нейронной сетью из множества обучающих примеров. Каждый бизнесмен и его предполагаемый бизнес характеризуются своим комплексом входных параметров, и поэтому вероятность успешности его бизнеса в каждом конкретном случае полезно оценивать путем вычислений с помощью нейронной сети — интеллектуальной системы прогнозирования успешности предпринимательской деятельности, разработанной авторами работы [84]. Обратиться к этой системе и проверить свою склонность к предпринимательской деятельности, а также выбрать наиболее предпочтительную сферу деятельности можно с сайта **www.PermAI.ru** (раздел «Проекты»).

Надо отметить, что данная интеллектуальная система имеет широкий спектр применения. Она может быть полезной для начинающих предпринимателей, а также служить в качестве инструмента поддержки решений, принимаемых зрелыми бизнесменами и экспертами инвестиционных компаний. И конечно же, она неоценима для молодых людей, решающих вопрос о выборе будущей профессии.

5.8. ВЫЯВЛЕНИЕ СПОСОБНОСТИ ЧЕЛОВЕКА К НАУЧНОЙ ДЕЯТЕЛЬНОСТИ

Как отмечалось в предыдущем параграфе, способности к бизнесу обнаруживаются не более чем у 7–10% населения. Считается, что научной деятельностью способно заниматься еще меньше людей.

В работе [3] нейросетевая система, прогнозирующая склонность к научной деятельности, создавалась путем обучения на биографических данных выдающихся ученых XIX—XX вв. Во внимание прини-

мались такие параметры, как: пол; национальность матери; национальность отца; этнолингвистическая группа; были ли в семье люди, связанные с наукой; был ли вундеркиндом в раннем возрасте; по каким предметам учился на 4 и 5; занимался ли под руководством выдающегося ученого; разносторонность интересов; политическая активность; рассеянность и неприспособленность к реальной жизни; проблемы, связанные с душевными расстройствами; отрасль науки.

Выходной параметр модели d кодировал результат прогноза: 1 — способен к научной деятельности, 0 — не способен к научной деятельности.

Всего было подготовлено множество из 200 примеров: 100 примеров со значением выходного параметра $d = 1$ и 100 примеров со значением выходного параметра $d = 0$.

Как уже отмечалось, при формировании примеров с выходным параметром $d = 1$ использовались данные из сети Интернет о выдающихся ученых XIX—XX вв., таких как В. Рентген, И. И. Мечников, П. Кюри, К. Ф. Гаусс и др. С примерами же, в которых $d = 0$, возникли проблемы, связанные с отсутствием информации. В связи с этим было введено предположение, что количество людей, имеющих способности к научной деятельности, значительно меньше количества людей, не имеющих таких способностей. На основании этого при формировании примеров, в которых $d = 0$, входные параметры генерировались случайным образом, однако с учетом известных статистических данных: каждый червертый-пятый человек в мире страдает психическими расстройствами; вундеркинды составляют 1% населения; на 100 девочек приходится 107 мальчиков и др. После удаления выбросов по методике [57] (см. § 6.5), появление которых, по-видимому, обусловлено недостоверностью информации, а также грубостью введенных гипотез, погрешность тестирования удалось снизить до 18%.

Как видно из рис. 5.9, исследования предметной области, выполненные с помощью нейросетевой модели, показали, что наиболее значимыми параметрами, определяющими способность человека к научной деятельности, являются:

1) занимался ли под руководством известного ученого;
2) был ли вундеркиндом в раннем возрасте;
3) был ли разносторонне развитым человеком;
4) был ли рассеянным и неприспособленным к жизни;
5) были ли в семье люди, связанные с наукой.

На рисунках 5.10 и 5.11 приведены результаты исследований влияния изменения входных параметров на результат моделирования — способность к научной деятельности. Исследование выполнено путем «замораживания» (фиксации) одних и постепенного изменения других входных параметров при одновременным наблюдении за значением

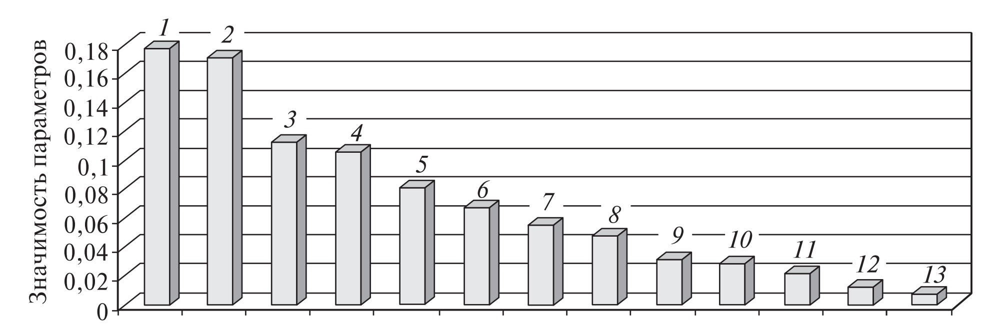


Рис. 5.9. Распределение значимости параметров, от которых зависит способность человека к научной деятельности

1 — занимался ли под руководством известного ученого; *2* — был ли вундеркиндом в раннем возрасте; *3* — был ли разносторонне развитым человеком; *4* — был ли рассеянным и неприспособленным к жизни; *5* — были ли в семье люди, связанные с наукой; *6* — был ли политически активным человеком; *7* — национальность отца; *8* — пол ученого; *9* — по каким предметам учился на 4 и 5; *10* — национальность матери; *11* — отрасль научной деятельности; *12* — этнолингвистическая группа; *13* — были ли проблемы, связанные с душевными расстройствами

выходного сигнала сети. Виртуальные эксперименты проводились над двумя учеными: А. Эйнштейном и И. Павловым.

Анализируя данные полученных в результате вычислительных экспериментов гистограмм можно отметить ряд интересных результатов.

1. Если бы Эйнштейн и Павлов не имели в качестве своих учителей известных ученых, у них было бы значительно меньше шансов стать знаменитыми учеными.
2. Такой же эффект наблюдается, если бы оба ученых не были разносторонне развитыми людьми.
3. Если бы Эйнштейн не был рассеянным и неприспособленным к жизни человеком, то это не повлияло бы на его способности к научной деятельности. Если бы Павлов был рассеянным и неприспособленным к жизни человеком, то его шансы стать знаменитым ученым упали бы до 4%.
4. Как известно, Эйнштейн и Павлов не были вундеркиндами в раннем возрасте. Если бы это было не так, т. е. если бы в раннем возрасте они были вундеркиндами, то их способности к научной работе остались бы по-прежнему высокими.
5. Известно, что в семьях, в которых воспитывались Эйнштейн и Павлов, не было людей, связанных с наукой. Если бы это было не так, то их способности к научной работе по-прежнему остались высокими.
6. Как известно, Эйнштейн активно занимался политикой, тогда как Павлов высокой политической активности не проявлял.

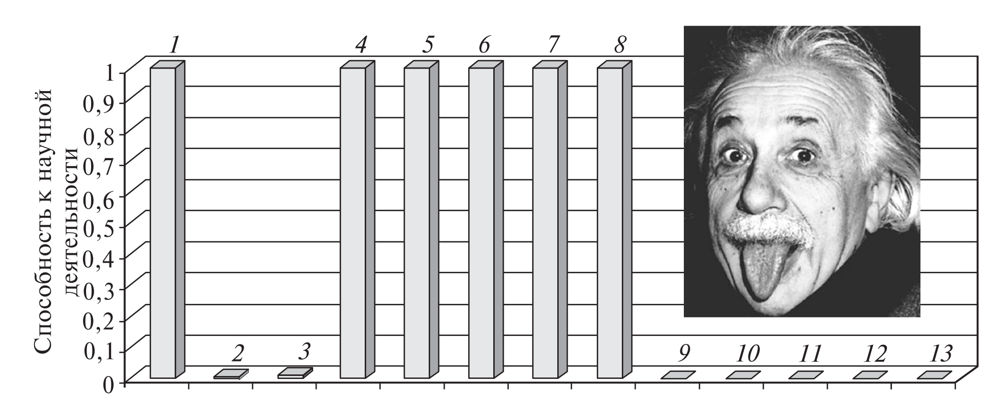


Рис. 5.10. Результаты компьютерных экспериментов, выполненных при виртуальном изменении социальных, биологических и биографических данных А. Эйнштейна

1 — исходное состояние; *2* — если бы не занимался под руководством известного ученого; *3* — если бы не было разносторонности интересов; *4* — если бы не был рассеянным и неприспособленным к жизни; *5* — если бы был вундеркиндом в раннем возрасте; *6* — если бы в семье были люди, связанные с наукой; *7* — если бы не был политактивным; *8* — если бы отец был славянином или армянином; *9* — если бы отец был французом, бельгийцем, англичанином, итальянцем, голландцем; *10* — если бы мать была англичанкой, латышкой, венгеркой, итальянкой, голландкой; *11* — если бы был женщиной; *12* — если бы был необучаем в детстве; *13* — если бы работал в области химии, физиологии, биологии, во многих областях

Компьютерный эксперимент показывают, что если бы было наоборот, т. е. если бы Павлов был политически активен, а Эйнштейн — нет, то на их способности к научной деятельности это бы не повлияло.

7. Если бы отец Эйнштейна был не евреем, а славянином или армянином, то его способности к научной работе остались бы по-прежнему высокими. Если бы его отец был французом, бельгийцем, англичанином, итальянцем или голландцем, то его способности к научной работе упали бы до нуля.
8. Если бы мать Эйнштейна была не немкой, а англичанкой, латышкой, венгеркой, итальянкой или голландкой, то его способности к научной работе упали бы до нуля.
9. Если бы отец или мать Павлова были не русскими, а любой другой национальности, то способности Павлова к научной работе остались бы по-прежнему высокими.
10. Если бы Эйнштейн был женщиной, имеющей такие же характеристики, то у него бы не было способностей к научной работе. Если бы Павлов был женщиной, то его способности к научной работе были бы на 7% меньше.

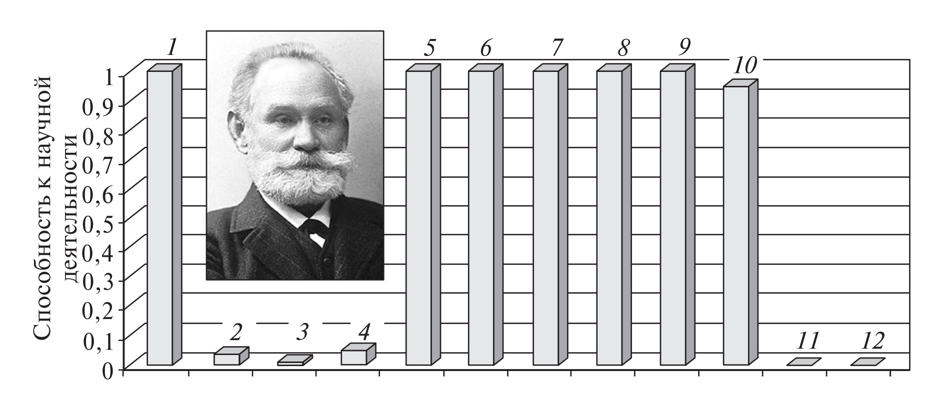


Рис. 5.11. Результаты компьютерных экспериментов, выполненных при виртуальном изменении социальных, биологических и биографических данных И. Павлова

1 — исходное состояние; *2* — если бы не занимался под руководством известного ученого; *3* — если бы не было разносторонности интересов; *4* — если бы был рассеянным и неприспособленным к жизни; *5* — если бы был вундеркиндом в раннем возрасте; *6* — если бы в семье были люди, связанные с наукой; *7* — если бы был политактивным; *8* — если бы отец был любой другой национальности; *9* — если бы мать была любой другой национальности; *10* — если бы был женщиной; *11* — если бы был необучаем в детстве; *12* — если бы работал в области математики, физики, химии, во многих областях

11. Если бы Эйнштейн и Павлов были неспособны обучаться в детстве, то у них бы не было способностей к научной работе.
12. Если бы Эйнштейн вместо физики работал в области химии, физиологии, биологии или во многих других областях, его шансы стать знаменитым ученым снизились бы до нуля. Если бы Павлов вместо физиологии работал в области математики, физики, химии или во многих других областях, то его шансы стать знаменитым ученым упали бы до нуля.

Подводя итог выполненным исследованиям, отметим, что приведенные здесь выводы на первый взгляд кажутся бесполезными и сомнительными. Но, во-первых, напомним, что они являются результатом компьютерных экспериментов над математической моделью, адекватность которой доказана сравнительно низкой погрешностью, полученной на тестовых примерах. А во-вторых, компьютерная программа, реализующая математическую модель, может быть реально использована как инструмент психофизиологической диагностики и профориентации. Например, молодым людям, мечтающим заниматься наукой, можно подбирать наиболее подходящую для них научную область, в которой их способности могут раскрыться наиболее эффективным образом. Воспользоваться таким сервисом можно в разделе «Проекты» сайта **www.PermAI.ru**.

5.9. СВОЙСТВА НЕЙРОННЫХ СЕТЕЙ, УНАСЛЕДОВАННЫЕ ОТ МОЗГА

Нейронная сеть — это система, выполненная по образу и подобию человеческого мозга. Вполне естественно, что она «наследует» его свойства, причем не только положительные, но и отрицательные.

Как и человеческий мозг, нейронная сеть способна получать знания из данных и обнаруживать скрытые в них закономерности. Но, как и человек, нейронная сеть не в состоянии объяснять, как она это делает?

Архимед открыл свой закон, лежа в ванне, Ньютон — наблюдая за падением яблока. Оба ученых гениальным образом догадались, нашли истину, не объяснив, однако, как они это сделали.

Нейронная сеть, извлекая закономерности из данных, тоже способна получать знания, делать догадки, открывать законы природы. Но так же, как и человек, нейронная сеть не способна к четкой формулировке пунктов алгоритма, позволившего ей сделать то или иное умозаключение. Она не способна ***вербализовать*** (от французского *verbal* — «словесный») свои выводы, т. е. объяснить их словами. Знания, которыми она обладает, ***невербальны***, т. е. необъяснимы посредством слов.

Согласно современным эзотерическим теориям, гений в момент озарения «считывает информацию из параллельных миров», поэтому он не может выстроить логическую цепочку, не может объяснить, как он догадался. Разумеется, нелепо было бы утверждать, что нейронная сеть в момент извлечения знаний проделывает аналогичную операцию, поэтому моделирование деятельности мозга дает нам основания возразить против эзотерической теории познания. По-видимому, многие из феноменов, которые демонстрируют экстрасенсы, контактеры или шаманы, можно объяснить свойством мозга извлекать информацию из неполных данных — свойством, которое удается моделировать с помощью нейронных сетей.

В практике применения нейронных сетей известны случаи, когда нейронные сети демонстрируют феномен, называемый в жизни «шестым чувством». ***Шестое чувство,*** или ***интуиция***, — это способность человека делать правильные выводы и принимать правильные решения, руководствуясь информацией, которой для логического объяснения таких выводов и решений недостаточно. Обычно считается, что «шестым чувством» в большей степени обладает женская половина человечества: очень часто женщины совершают поступки и принимают решения, которые мужчины (с точки зрения их традиционной логики) объяснить не могут, — и нередко такие поступки и решения оказываются единственно правильными.

Нейронные сети с успехом извлекают знания при анализе информации, из которой, казалось бы, эти знания извлечь невозможно. Ранее, в § 5.3, был приведен пример диагностики неисправностей авиационных двигателей по совокупности их полетных параметров. Феномен заключается в том, что, помимо всех прочих неисправностей, нейронная сеть диагностирует и такие, которые невозможно выявить традиционными диагностическими методами, основанными на законах физики. Так, дефект «стружка в масле» обнаруживается только после вскрытия авиадвигателя. Этот дефект не выявляется традиционными диагностическими методами, поскольку его наличие, по мнению специалистов, никак не влияет ни на один из снимаемых с авиадвигателя полетных параметров. Тем не менее, несмотря на отсутствие какой-либо связи между появлением этого дефекта и параметрами работы авиадвигателя, нейронная сеть все-таки обнаруживает скрытую от обычных вербальных методов диагностики закономерность и ставит правильный диагноз относительно наличия или отсутствия стружки.

Как можно объяснить этот феномен?

Наше объяснение состоит в следующим. Действительно, согласно общепринятому мнению специалистов, появление стружки в поддоне двигателя не сказывается на его температуре, вибрациях, шумах, перепадах давления и пр., а следовательно, не вызывает серьезных отклонений ни одного из значений параметров входного вектора $\boldsymbol{X}$: $x_1, x_2, \ldots$. Но возможно, что небольшие, незаметные для глаза специалистов отклонения в параметрах работы двигателя все-таки есть. Причем, возможно, появление стружки приводит к изменениям не одного и не двух, а сразу многих компонент вектора $\boldsymbol{X}$, и нейросеть реагирует на этот комплекс незаметных глазу изменений. Ведь она «следит» не за одним и не за двумя, а сразу за всеми компонентами вектора $\boldsymbol{X}$!

Феномены подобного рода в практике применения нейросетевых технологий не являются редкостью. Особенно часто они наблюдаются при исследовании достаточно сложных объектов, когда нейронные сети выявляют связи и закономерности, о существовании которых специалисты не знают. Выявляются даже такие взаимосвязи и взаимные влияния, которые при поверхностном рассмотрении противоречат здравому смыслу специалистов и становятся понятными и объяснимыми только после более тщательного изучения объекта.

Способность нейронных сетей и нейрокомпьютеров выявлять и использовать неизвестные ранее знания — это, безусловно, их положительное свойство, «унаследованное» ими от своего естественного прототипа — мозга. Мы уже были свидетелями тому, как это свойство позволило создать нейросетевую систему диагностики авиационных двигателей, способную выявлять дефекты, которые традиционными методами не выявляются. В других случаях обнаружение нейросетевыми моделями новых, неизвестных ранее знаний и закономерностей

предметных областей может приводить к открытиям, изменяющим традиционные научные представления и парадигмы, порождающим инновационные прорывы в той или иной научной области. Так, исследование нейросетевой кардио-диагностико-прогностической системы (см. § 5.2 и [69, 108, 109]) позволило сделать принципиально новый вывод о том, что общие рекомендации, обычно даваемые врачами всем без исключения кардиологическим больным о необходимости соблюдения гипохолестериновой диеты, отказа от употребления кофе и алкоголя, снижения веса, ограничения физических нагрузок и др., для некоторых пациентов могут оказаться не только не полезными, но и причинить вред. Исследования [108, 109] показали, что такие рекомендации должны даваться с учетом индивидуальных особенностей больных, определяемых довольно большим комплексом их параметров.

В отличие от экспертных систем, в которые знания закладываются экспертами-разработчиками, нейронные сети приобретают знания автоматически, извлекая их из примеров практического поведения предметных областей. И здесь уместно вспомнить слова Гете: «Суха теория, мой друг, но древо жизни зеленеет». Всякая модель мира всегда ограничена. Всякая теория беднее практики. Поэтому нейронные сети, обучающиеся непосредственно на практических примерах реальной жизни, в принципе способны извлекать и усваивать такие знания, которые специалистам-экспертам и ученым, а значит и создателям экспертных систем, не известны.

Более того, они способны извлекать и использовать и такие знания, которые на современном этапе развития традиционной науки объяснить не удается. Например, повышение точности прогнозов курса валют в случае учета факторов космической природы [71] или влияние цвета глаз предпринимателя на успешность его бизнеса (§ 5.7 и [84]).

Итак, *невербальность* и *интуиция* нейронных сетей — это свойства, «унаследованные» ими от мозга. Причем если первое из этих свойств является вредным, то второе — полезным. ***Невербальность*** вызывает трудности в применении нейронных сетей и осложняет нахождение общего языка с потребителями программной продукции, которым часто требуется не только получить какой-либо вывод, но и объяснить, на основании чего он сделан. Свойство ***интуиции***, наоборот, делает их одним из эффективнейших инструментов, полезных для научного и практического применения.

Интересно отметить, что кроме вышеназванных обнаружились и другие свойства нейронных сетей, тоже унаследованные от мозга. Это, например, *свойство живучести*. Известно, что мозг очень живуч: были случаи, когда люди, потеряв в результате травм до 30% объема мозга, вели себя вполне адекватно и продолжали совершать обдуманные поступки. Как выяснилось, нейрокомпьютеры (в отличие от обычных фон-неймановских компьютеров) тоже продолжают

функционировать при довольно значительных повреждениях! Они продолжают принимать правильные решения в случае нарушения одного или нескольких внутренних соединений — синаптических связей. Они продолжают решать поставленные перед ними задачи даже в случае удаления из них некоторого количества нейронов.

Еще одним свойством, унаследованным от мозга, по-видимому, является свойство *гиперразмерности* (или *переобучения*), под которым понимается ухудшение обобщающих свойств нейросети при излишне большом количестве скрытых нейронов (см. § 6.1). В качестве аналогии из реальной жизни можно привести известное наблюдение психологов того, что эрудиция и креативность редко уживаются в одном человеке, что эти качества людей часто являются взаимоисключающими. Еще один пример могут привести опытные педагоги, знающие, что в реальной жизни троечники обычно устраиваются лучше, чем отличники.

Итак, персептрон, являясь моделью человеческого мозга, наследует его свойства, причем не только положительные, но и отрицательные.

К положительным относятся следующие свойства:

- *обучение*;
- *обобщение*;
- *извлечение знаний из данных*;
- *интуиция*;
- *живучесть*.

Отрицательными качествами являются *невербальность* и *гиперразмерность*. Заметим, что если свойство обучения первоначально планировалось первыми создателями нейрокомпьютеров, то остальные обнаружились в дальнейшем по мере эксплуатации, развития и изучения нейронных сетей и нейрокомпьютеров.

Контрольные вопросы и задания к § 5.9

1. Перечислите положительные и отрицательные свойства, которые персептрон наследовал от своего прототипа — человеческого мозга.
2. Какие знания называются невербальными?
3. Вспомните, каким образом вербализуются выводы, получаемые с помощью экспертных систем.
4. Подумайте над тем, как можно вербализовать нейросетевые знания.
5. Дайте определение интуиции. Приведите примеры проявления этого свойства у человека и у нейронных сетей.
6. В каких случаях может быть полезно свойство живучести нейрокомпьютеров?
7. Приведите примеры выявления с помощью метода нейросетевого моделирования закономерностей, которые на современном этапе развития традиционной науки объяснить не удается. Попытайтесь дать объяснение этим закономерностям.

5.10. КРУГ ЗАДАЧ, РЕШАЕМЫХ ПРИ ПОМОЩИ НЕЙРОННЫХ СЕТЕЙ

Приведенные выше примеры, а также анализ литературных данных и интернет-источников убедительно показывают, что нейросетевые технологии — это универсальный и весьма эффективный инструмент для построения математических моделей и интеллектуального анализа данных в самых разнообразных физических, технических, химических, экономических, социальных и другого рода объектах, процессах и явлениях. Проводя исследование предметных областей с помощью этих моделей, мы можем решать разнообразные практические задачи. Например, если нам удалось построить математическую модель какого-то сложного технологического процесса (скажем, выплавки стали в электродуговой печи, или крекинга нефти в химическом реакторе, или производства электроэнергии на электростанции), то, исследуя эту компьютерную математическую модель, проводя над ней *виртуальные эксперименты*, изучая с их помощью влияние входных параметров на выходные, можно решить *задачу оптимизации* моделируемого технологического процесса. Это значит, что можно подобрать оптимальное сочетание входных параметров, обеспечивающих, например, максимально высокое качество выплавляемой стали; можно рассчитать наиболее благоприятный ход химической реакции крекинга нефти; можно выбрать наиболее эффективный режим работы электростанции и т. д.

Аналогично решаются задачи оптимизации в сфере бизнеса и экономики. В этом случае выходом нейронной сети может быть некая *целевая функция*, имеющая смысл экономической эффективности предприятия: валового продукта, прибыли или рентабельности фирмы.

Если математическая модель является *нестационарной*, т. е. составлена с учетом фактора времени, то ее можно использовать для решения *задач прогнозирования*. Это значит, что с помощью математической модели можно узнать, какими будут технологические, экономические, социальные, политические, спортивные, медицинские, экологические и другие показатели моделируемого объекта или процесса в будущем и как на них можно повлиять, принимая те или иные меры сегодня.

Если математическая модель работает в реальном режиме времени, т. е. оперативно получает сведения о текущих изменениях параметров моделируемого объекта, если результаты математического моделирования могут быть переданы оператору, управляющему объектом, или непосредственно введены в приборы, дозирующие подачу руды, кислорода и других химических компонентов в электропечь либо в приборы, управляющие параметрами работы химического или ядерного реактора, то такая математическая модель будет решать *задачу управления* моделируемым объектом или процессом.

Наконец, если нейросеть каким-либо образом вербализовать, то ее можно использовать для *понимания* явлений и процессов, происходящих в изучаемой предметной области.

Кроме перечисленных задач — понимания, оптимизации, прогнозирования и управления — персептрон может решать задачи *распознавания* и *классификации образов*. Под образами здесь понимаются зрительные изображения, символы, тексты, запахи, звуки, шумы, наборы параметров (векторы), характеризующие состояние здоровья пациента или состояние технического устройства и пр.

Отметим, что во всех ранее рассмотренных примерах построения нейросетевых математических моделей не требовалось знание и использование законов природы, экономики, общества. Вместо этого нужно было лишь назначить входные и выходные параметры создаваемой модели и подготовить множество примеров, содержащее статистические данные о поведении предметной области. И если предметная область такова, что между входными и выходными параметрами модели существуют взаимозависимости, а множество примеров достаточно *репрезентативно* (представительно), то нейронная сеть сама извлекает закономерности предметной области и кодирует их в виде сил синаптических связей так, что нейросетевая математическая модель становится адекватной по отношению к рассматриваемой предметной области.

Контрольные вопросы и задания к § 5.10

1. В чем состоит задача оптимизации моделируемого объекта или процесса и как она решается с помощью нейросетевых технологий? Приведите примеры.
2. В чем состоит задача прогнозирования свойств моделируемого объекта, явления или процесса и как она решается с помощью нейросетевых технологий? Приведите примеры.
3. В чем состоит задача управления моделируемым объектом или процессом и как она решается с помощью нейросетевых технологий? Приведите примеры.
4. В чем состоит задача распознавания (классификации) образов и как она решается с помощью нейросетевых технологий? Приведите примеры.
5. Чем отличается искусственная нейронная сеть от нейрокомпьютера?

Глава 6

ОПТИМАЛЬНОЕ ПРОЕКТИРОВАНИЕ И ОБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ

6.1. ТЕОРЕМЫ СУЩЕСТВОВАНИЯ

Из предыдущего изложения следуют два важных вывода.

1. В жизни встречается множество практически важных задач, решить которые можно методом математического моделирования, т. е. путем построения некоторой сложной функции, осуществляющей преобразование вектора входных параметров $\boldsymbol{X}$ в вектор выходных параметров $\boldsymbol{D}$.

2. Универсальным инструментом построения такой функции являются нейросетевые технологии.

Естественно, возникают вопросы: всегда ли можно построить нейронную сеть, выполняющую преобразование, заданное любым множеством обучающих примеров, и каким требованиям должна удовлетворять эта нейронная сеть?

Надо признать, что до сих пор материал, который мы изучали, носил в большей степени позитивный, «рекламный» характер. Мы много говорили о преимуществах и широких возможностях метода нейросетевого моделирования, но практически почти ничего — о его недостатках. Мы не упоминали о трудностях, связанных с применением этого метода. Но освоить любой метод можно, только познакомившись со всеми его проблемами, изучив способы их преодоления, изучив и поняв его теоретическую базу.

С одной из таких трудностей, названной «Проблемой исключающего ИЛИ», мы уже столкнулись при изучении § 4.6 и теперь знаем, что персептрон должен иметь скрытый слой нейронов, но осталось два вопроса.

1. Всегда ли можно спроектировать и обучить многослойный персептрон, обеспечивающий решение любой задачи?

2. Каким образом лучше задавать количество внутренних нейронных слоев и количество нейронов в них? Может быть, как в мозге, — 10^{11} нейронов? Может, чем их будет больше, тем лучше?

Ответы на эти вопросы мы выясним, познакомившись с *теоретической базой нейронных сетей*.

Важнейшее место в теории нейронных сетей занимает *теорема Арнольда—Колмогорова—Хехт-Нильсена*, доказательство которой до-

статочно сложно и поэтому в нашем курсе не рассматривается. Тем не менее сама история этой теоремы представляется весьма интересной и поучительной.

Как мы уже не раз отмечали, с физической точки зрения персептрон — это устройство, моделирующее человеческий мозг на структурном уровне. Однако, анализируя формулы, по которым он преобразует сигналы, можно заметить, что с математической точки зрения персептрон — это всего лишь *аппроксиматор*, заменяющий функцию многих аргументов суммой функций, каждая из которых зависит только от одного аргумента. Вопрос о том, всегда ли можно любую функцию многих аргументов представить в виде суммы функций меньшего количества аргументов, интересовал математиков на протяжении нескольких столетий.

Так, в 1900 г. на Всемирном математическом конгрессе в Париже знаменитый немецкий математик Давид Гильберт сформулировал 23 проблемы, которые он предложил решать математикам начинающегося XX в. Одна из этих проблем (под номером 13) как раз и декларировала невозможность такого представления. Таким образом, приговор новой области искусственного интеллекта был вынесен за полвека до ее появления. Получалось, что персептрон, сколько бы нейронов он ни имел, не всегда мог построить нужную математическую функцию.

Многие проблемы Гильберта были успешно решены математиками XX в. Его утверждения-гипотезы подтверждались одна за другой, но вот с тринадцатой возникли проблемы.

Сомнения относительно возможностей персептронов развеяли советские математики — академики В. И. Арнольд [2, 66] и А. Н. Колмогоров [27, 28]. Им удалось доказать, что любая непрерывная функция n аргументов $f(x_1, x_2, \ldots, x_n)$ всегда может быть представлена в виде суммы непрерывных функций одного аргумента: $f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)$. Тем самым гипотеза Гильберта была опровергнута, а нейроинформатике был открыт «зеленый свет».

В 1987–1991 гг. профессор Калифорнийского университета (США) Р. Хехт-Нильсен [95, 96] переработал теорему Арнольда — Комогорова применительно к нейронным сетям. Он доказал, что ***для любого множества различающихся между собой пар векторов $\boldsymbol{X}_q$ и $\boldsymbol{D}_q$ произвольной размерности существует двухслойный персептрон с сигмоидными активационными функциями и с конечным числом нейронов, который для каждого входного вектора $\boldsymbol{X}_q$ формирует соответствующий ему выходной вектор $\boldsymbol{D}_q$***. Таким образом, была доказана принципиальная возможность построения нейронной сети, выполняющей преобразование, заданное *любым* множеством различающихся между собой обучающих примеров, и установлено, что такой универсальной нейронной сетью является двухслойный персептрон —

персептрон с одним скрытым слоем, причем активационные функции его нейронов должны быть сигмоидными.

Теорема Арнольда — Колмогорова — Хехт-Нильсена имеет очень важное для практики следствие в виде формулы, с помощью которой можно определять необходимое количество синаптических весов нейронной сети:

$$\frac{N_y Q}{1 + \log_2(Q)} \leqslant N_w \leqslant N_y \left(\frac{Q}{N_x} + 1 \right) \left(N_x + N_y + 1 \right) + N_y, \tag{6.1}$$

где N_x — количество нейронов входного слоя; N_y — количество нейронов выходного слоя; Q — количество элементов множества обучающих примеров, т. е. количество пар входных и выходных векторов $\boldsymbol{X}_q$ и $\boldsymbol{D}_q$; N_w — необходимое число синаптических связей.

Оценив с помощью этой формулы необходимое число синаптических связей N_w, можно рассчитать и необходимое количество нейронов в скрытых слоях. Например, количество нейронов скрытого слоя двухслойного персептрона будет равно:

$$N = \frac{N_w}{N_x + N_y}\,. \tag{6.2}$$

Последняя формула становится очевидной, если ее левую и правую части умножить на $(N_x + N_y)$ и нарисовать схему двухслойного персептрона (т. е. персептрона с одним скрытым слоем).

А теперь зададимся вопросом: что было бы, если бы гипотеза 13-й проблемы Гильберта оказалась верной? Очевидно, тогда у нейроинформатики не было бы теоретического фундамента. Ее прикладные возможности были бы серьезно ограничены; более того, сама правомерность ее существования как области науки оказалась бы под вопросом! Кому нужен нейрокомпьютер, который в принципе не может решать ряд задач, для решения которых он предназначен? Не было бы у нейроинформатики ее многочисленных практических приложений…

Но они есть! В настоящее время нейроинформатику можно заносить в Книгу рекордов Гиннеса: она побила все рекорды по количеству приложений в самых разнообразных областях! Жизнь сама блестяще подтвердила теоремы и доказательства, выполненные в 1956–57 гг. нашими соотечественниками — Андреем Николаевичем Колмогоровым и Владимиром Игоревичем Арнольдом!

Между тем среди математиков не прекращаются попытки развития и усиления теорем Арнольда — Колмогорова — Хехт-Нильсена. Так, известным российским ученым А. Н. Горбанем доказано [16, 17], что универсальные аппроксимационные свойства нейросетей определяются не конкретным выбором функций активации (ими не обязательно должны быть сигмоиды), а свойством нелинейности этих функций:

«достаточно, чтобы активационные функции были нелинейны и дважды непрерывно дифференцируемы».

У нас нет оснований сомневаться в строгости математических доказательств профессора А. Н. Горбаня, но практика применения нейросетевых технологий убедительно показывает, что наилучшие нейросетевые модели (персептронного типа) получаются в тех случаях, когда в качестве активационных функций используются функции именно сигмоидного типа — те, которые зафиксированы биологами при изучении мозга, т. е., которые придумали не математики, а сама Природа.

Хотя возможны и исключения.

6.2. МЕТОДЫ ПРОЕКТИРОВАНИЯ НЕЙРОННЫХ СЕТЕЙ

Как следует из теорем Арнольда — Колмогорова — Хехт-Нильсена, для построения нейросетевой модели любого сколь угодно сложного объекта достаточно использовать персептрон с одним скрытым слоем сигмоидных нейронов, количество которых определяется формулами (6.1), (6.2). Однако в практических реализациях персептронов оптимальное количество как слоев, так и нейронов в каждом из них нередко отличается от теоретических. К тому же иногда бывает целесообразно использовать персептроны с бо́льшим количеством скрытых слоев.

Строгой теории выбора оптимального количества скрытых слоев и нейронов в скрытых слоях пока не существует. На практике чаще всего используются персептроны, имеющие один или два скрытых слоя, причем количество нейронов в скрытых слоях обычно колеблется от $N_x/2$ до $3N_x$.

При проектировании персептрона необходимо понимать, что он должен не только правильно реагировать на примеры, на которых он обучен, но и уметь ***обобщать*** приобретенные знания, т. е. правильно реагировать на примеры, которых в обучающем множестве не было. Чтобы оценить способность сети к обобщению, помимо ***обучающего множества*** примеров $\boldsymbol{L}$, в рассмотрение вводят некоторое количество тестовых примеров, образующих ***тестирующее множество*** $\boldsymbol{T}$ (рис. 6.1). Причем примеры тестирующего множества относятся к той же самой предметной области, но в процессе обучения не участвуют.

После обучения вычисляют среднеквадратичную (или максимальную) ошибку нейронной сети как разницу между получившимися в результате вычислений компонентами выходного вектора $\boldsymbol{Y}$ и его желаемыми (заданными в примерах) компонентами вектора $\boldsymbol{D}$. Ошибку, вычисленную на примерах обучающего множества $\boldsymbol{L}$, называют ***ошибкой обучения*** и обозначают как ε_L, а ошибку, вычисленную на примерах тестирующего множества $\boldsymbol{T}$, называют ***ошибкой обобщения***, или ***ошибкой тестирования***, и обозначают как ε_T.

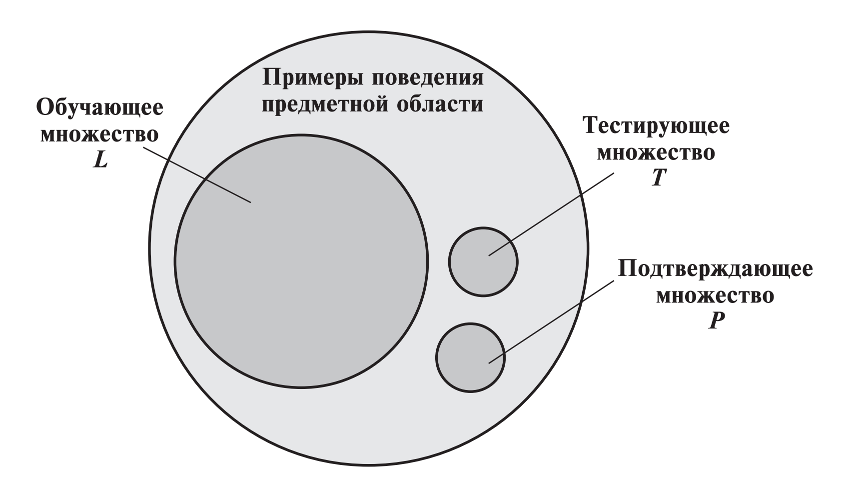


Рис. 6.1. Деление примеров поведения предметной области на обучающее множество ***L***, тестирующее множество ***T*** и подтверждающее множество ***P***

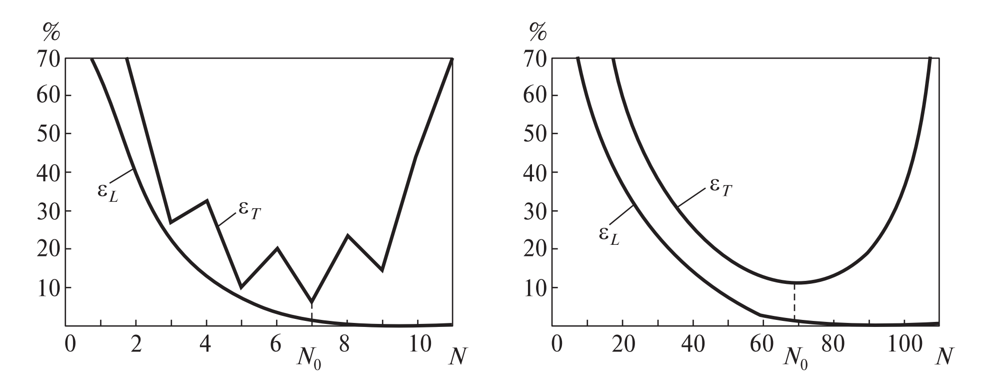


Рис. 6.2. Характерные зависимости ошибки обучения ε_L и ошибки обобщения (тестирования) ε_T от количества скрытых нейронов N

Обратим внимание, что ошибка обучения ε_L вычисляется на тех примерах предметной области, на которых сеть обучалась. Ошибка же обобщения ε_T вычисляется на примерах той же предметной области, но которые сеть никогда «не видела». Поэтому ошибки ε_L и ε_T имеют разную природу и, соответственно, разный характер поведения. При увеличении количества N нейронов скрытых слоев персептрона ошибка обучения ε_L обычно уменьшается, тогда как ошибка обобщения ε_T сначала уменьшается, а затем (начиная с некоторого оптимального количества нейронов скрытого слоя $N = N_0$) вновь возрастает. Характерные кривые зависимости ошибок обучения и обобщения от количества скрытых нейронов приведены на рис. 6.2.



Рис. 6.3. Аппроксимация данных предметной области с помощью полиномов первого порядка (кривая *1*), второго порядка (кривая *2*) и пятого порядка (кривая *3*)

Поведение этих кривых легко объяснить, если воспользоваться аналогией с аппроксимацией набора данных полиномами методом наименьших квадратов. Как известно, задача аппроксимации состоит в том, чтобы подобрать полином, наиболее правильно отражающий характер закономерности, представленной графически точками предметной области. На рис. 6.3 точки, соответствующие некоторым параметрам предметной области, изображены в системе координат $x - y$ в виде двенадцати черных и белых кружков. Точки, отмеченные черными кружками, использовались при определении коэффициентов аппроксимирующих полиномов, а точки, отмеченные белыми кружками, не использовались, поэтому по ним можно судить о качестве аппроксимации. Выражаясь терминами нейросетевых технологий, черные кружки можно назвать обучающими примерами, а белые — тестирующими.

Кривая *1* изображает результат аппроксимации полиномом первой степени, кривая *2* — полиномом второй степени, кривая *3* — полиномом пятой степени. Как видно из рисунка, использование полинома первой степени не дает хорошей аппроксимации закономерности предметной области. Кривая, соответствующая полиному пятой степени, в точности проходит через все черные кружки, т. е. имеет нулевую погрешность обучения ε_L, тогда как тестовые белые кружки остаются в стороне, и значит, погрешность тестирования ε_T весьма велика. Следовательно, данная кривая свойством обобщения не обладает. Наименьшую погрешность тестирования ε_T имеет кривая второго порядка, которая для данной предметной области является оптимальной.

Подводя итог проводимому сравнению, отметим, что нейросети, как и регрессионные модели, выстраивают функции, аппроксимирующие точки предметной области, причем для каждой решаемой задачи существует некоторое оптимальное значение степеней свободы математической модели. В случае полиномиальной аппроксимации степенями свободы являются определяемые коэффициенты (число которых на единицу больше порядка полинома). Число степеней свободы персептрона — это общее количество синаптических весов (и порогов), которое определяется количеством нейронов скрытых и выходных слоев. Свойство нейросети терять способность к обобщению при чрезмерном увеличении количества скрытых нейронов (степеней свободы) называют *переобучением*, или *гиперразмерностью*.

Вооружившись этими теоретическими знаниями, приведем некоторые практические рекомендации по проектированию персептронов, т. е. по выбору количества входных нейронов, количества выходных нейронов, количества скрытых слоев, количества нейронов в скрытых слоях, виду активационных функций.

1. Количество входных нейронов персептрона должно совпадать с количеством входных параметров, т. е. с размерностью вектора $\boldsymbol{X}$, который определен условиями решаемой задачи.

2. Количество нейронов выходного слоя должно совпадать с количеством выходных параметров, т. е. с размерностью выходного вектора $\boldsymbol{D}$, что также определено условиями задачи.

3. Количество скрытых слоев персептрона, согласно теоремам Арнольда — Колмогорова — Хехт-Нильсена, должно быть не менее одного, причем нейроны в скрытых слоях должны иметь сигмоидную активационную функцию.

4. Количество нейронов в скрытых слоях может быть приближенно оценено по формулам следствия теоремы Арнольда — Колмогорова — Хехт-Нильсена (6.1), (6.2). Однако его желательно уточнять (оптимизировать) для каждой конкретной задачи, например, путем построения кривых рис. 6.2 и определения с их помощью оптимального количества скрытых нейронов N_0.

5. Согласно теоремам Арнольда — Колмогорова — Хехт-Нильсена, активационные функции хотя бы одного скрытого слоя нейронов должны иметь форму сигмоиды.

В наиболее ответственных случаях наряду с обучающим множеством $\boldsymbol{L}$ и тестирующим множеством $\boldsymbol{T}$, в рассмотрение вводят еще одно множество примеров, называемое *подтверждающим*, и обозначают его буквой $\boldsymbol{P}$ (см. рис. 6.1). Примеры этого множества также относятся к рассматриваемой предметной области, но ни в обучении, ни в тестировании и оптимизации сети они не участвуют. Поэтому *ошибка прогнозирования* ε_P, вычисленная на примерах подтверждающего

множества $\boldsymbol{P}$, является наиболее объективной оценкой качества нейросетевой математической модели.

Для проектирования нейронных сетей и работы с ними имеется множество специальных программ, называемых *нейросимуляторами*, *нейроимитаторами*, *нейропакетами*. Многие из этих программ имеют функции автоматической оптимизации нейронных сетей.

Контрольные вопросы и задания к § 6.2

1. В чем состоит процесс проектирования персептронов?
2. Как задается количество нейронов входного слоя персептрона?
3. Как задается количество нейронов выходного слоя персептрона?
4. Как задается количество нейронов в скрытых слоях персептрона?
5. От каких параметров задачи зависит оптимальное количество нейронов скрытых слоев персептрона?
6. Как задается вид активационных функций нейронов?
7. Чем отличается ошибка обучения от ошибки тестирования?
8. Нарисуйте примерные графики зависимости обеих ошибок от количества нейронов скрытых слоев персептрона.
9. К какому нежелательному последствию может привести чрезмерное уменьшение количества нейронов в скрытых слоях персептрона?
10. К какому нежелательному последствию может привести чрезмерное увеличение нейронов в скрытых слоях персептрона?
11. Как на практике подбирается количество нейронов скрытых слоев персептрона?
12. Почему ошибка, вычисленная на примерах подтверждающего множества, является наиболее объективной оценкой качества нейронной сети?
13. Попробуйте объяснить с точки зрения теории нейронных сетей (см. § 6.1) часто встречающийся факт — «двоечники» и «троечники» в жизни устраиваются лучше, чем отличники.

6.3. МЕТОДЫ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Как было отмечено ранее, изобретение алгоритма обратного распространения ошибки открыло путь широкому практическому применению многослойного персептрона. Однако с расширением фронта научных исследований обнаружились и недостатки этого алгоритма.

Метод градиентного спуска. Прежде всего отметим, что алгоритм обратного распространения ошибки в его первоначальном изложении реализовывал *метод градиентного спуска*, который является далеко не самым лучшим методом теории оптимизации. Эта теория ставит своей задачей поиск минимума некоторой целевой функции (функционала) ε,

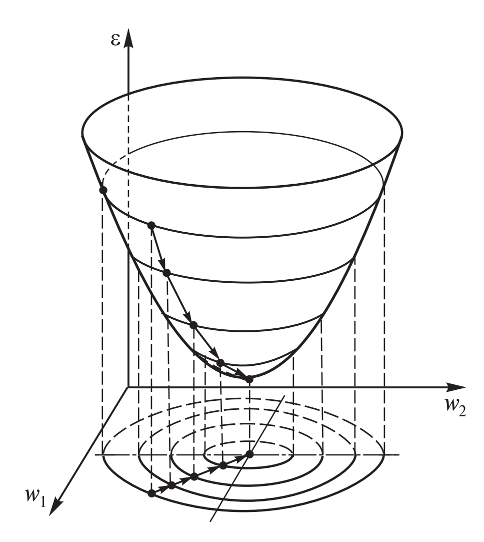


Рис. 6.4. Поиск минимума функции $\varepsilon = \varepsilon(\boldsymbol{w})$ методом градиентного спуска

которая зависит от нескольких переменных, представленных в виде вектора $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]^T$. В случае обучения многослойного персептрона целевая функция ε — это квадратичная ошибка персептрона, рассчитанная по формуле (4.14), а $\boldsymbol{w}$ — вектор синаптических весов межнейронных связей.

Согласно методу градиентного спуска, процесс поиска минимума функции $\varepsilon = \varepsilon(\boldsymbol{w})$ можно представить следующим образом. В некоторой случайно заданной точке поверхности ошибок (рис. 6.4) находится направление скорейшего спуска (антиградиента), затем делается шаг вниз на расстояние, пропорциональное крутизне склона (градиенту) и коэффициенту скорости обучения η. В новой точке снова определяются направление и величина антиградиента, согласно которому делается следующее перемещение и т. д.

Методы оптимизации, использующие понятие градиента функции, называются *градиентными*. Существует группа методов, называемых квазиньютоновскими, в которых, помимо первых производных, используются еще и вторые производные целевой функции, т. е., помимо градиента функции ошибок, учитывается еще и ее кривизна.

Во всех методах обучения нейросетей существует проблема выбора параметра η, определяющего длину шага вдоль выбранного направления оптимизации. Простейший способ состоит в фиксации постоянных значений η на весь период обучения. При этом рекомендуется величину η задавать отдельно для каждого слоя персептрона, например, по эмпирической формуле [41, 54]

$$\eta \leqslant \min\left(\frac{1}{n_i}\right), \tag{6.3}$$

где n_i — количество входов i-го нейрона в слое.

Другие методики предполагают динамическое изменение η в ходе обучения в зависимости от поведения целевой функции ошибок ε, вычисляемой по формуле (4.14). Для более быстрой сходимости коэффициент η стремятся увеличивать по мере снижения функции ошибок, однако не допуская ее существенного возрастания.

Эвристические методы. Следующую группу методов обучения нейросетей называют *эвристическими*. Эти методы не имеют строгого теоретического обоснования, но в них отражается личный опыт работы авторов в области нейросетевых технологий. В качестве примера приведем метод упругого обратного распространения (алгоритм RPROP — от *Resilent back PROPagation*). Согласно этому алгоритму, при корректировке весовых коэффициентов учитывается только знак соответствующей составляющей градиента, а само значение градиента в расчет не принимается:

$$\Delta w_{ij}(t) = -\eta_{ij}(t) \cdot \operatorname{sign}\left(S_{ij}(t)\right),$$

где $S_{ij}(t) = \dfrac{\partial \varepsilon(t)}{\partial w_{ij}}$

Коэффициенты обучения рассчитываются с помощью эвристической формулы

$$\eta_{ij}(t) = \begin{cases} 1{,}2\eta_{ij}(t-1), & \text{если } S_{ij}(t) \times S_{ij}(t-1) > 0; \\ 0{,}5\eta_{ij}(t-1), & \text{если } S_{ij}(t) \times S_{ij}(t-1) < 0; \\ \eta_{ij}(t-1), & \text{если } S_{ij}(t) \times S_{ij}(t-1) = 0. \end{cases}$$

Таким образом, идея метода, геометрически интерпретированная на рис. 6.5, состоит в том, что коэффициент обучения для каждого весового коэффициента увеличивается до тех пор, пока знак градиента на двух последующих шагах не сменится на противоположный. В случае же смены знака градиента коэффициент обучения уменьшается в 2 раза. Причем авторы алгоритма ограничили рост коэффициентов обучения числом $\eta_{\max} = 50$, а его уменьшение — числом $\eta_{\min} = 10^{-6}$.

Как видим, помимо основной идеи, которая понятна на интуитивном уровне, эвристический алгоритм RPROP содержит четыре эмпирических параметра.

Неградиентные методы. Существует группа методов, не требующих вычисления производных от целевой функции и поэтому называемых *неградиентными*. В таких методах используется только информация о значениях целевой функции в отдельных точках, на основании которой строится стратегия поиска экстремума. В качестве примеров можно назвать метод половинного деления, метод золотого сечения, метод Фибоначчи и др. Однако, как правило, неградиентные методы значительно уступают по эффективности градиентным и поэтому не рекомендуются для обучения нейросетей.

В настоящее время основная проблема обучения персептронов состоит в том, что поверхность функции ошибок обычно имеет очень сложную форму с множеством локальных минимумов (рис. 6.6). Поэтому все изложенные выше методы часто приводят к одному из локальных минимумов, лежащих в окрестности начальной точки обучения. Если после нахождения такого минимума погрешность обучения

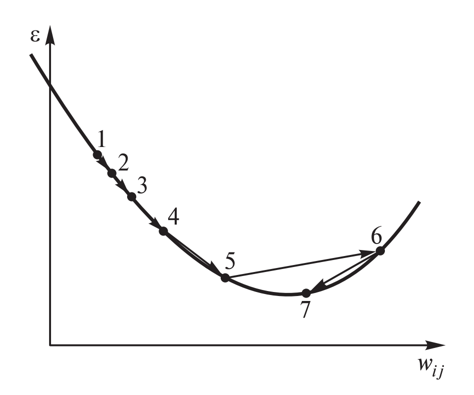


Рис. 6.5. Геометрическая интерпретация к алгоритму упругого обратного распространения RPROP

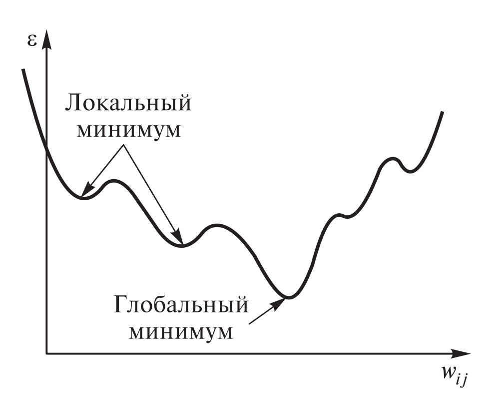


Рис. 6.6. Глобальный и локальные минимумы на поверхности ошибок

нейросети признается неудовлетворительной, то сеть «встряхивают», давая весовым коэффициентам случайные приращения, и продолжают процесс обучения из другой точки.

Часто процесс обучения приобретает характер длительного экспериментирования, в ходе которого пробуются различные оптимизационные алгоритмы с различными параметрами. В результате успех применения нейросетевых технологий ставится в зависимость от опыта и интуиции специалиста, количества и качества различных оптимизационных алгоритмов, имеющихся в его распоряжении. В связи с этим актуальным является развитие методов *глобальной оптимизации*, т. е. таких, которые позволяют находить глобальные экстремумы многоэкстремальных целевых функций. Среди множества возможных подходов на пути решения этой проблемы наиболее перспективными признаются *генетические алгоритмы*.

6.4. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Все живое, что нас окружает в этом мире, и мы сами удивительно совершенны. Мы имеем оптимальное количество рук, ног, пальцев, внутренних органов, живем в среднем оптимальное количество лет, имеем оптимальные пропорции, оптимальный химический состав, строение и т. д. Много веков люди пытаются найти и объяснить источник этого совершенства, понять механизм его возникновения. Существуют древние учения и религии, объясняющие первопричину совершенства и гармонию мира. Современному же человеку наиболее близка для восприятия эволюционная теория, основы которой были заложены Чарльзом Дарвином в 1859 г. в его знаменитой работе

«Происхождение видов путем естественного отбора». Согласно эволюционной теории, природа оптимизирует все живое благодаря двум биологическим механизмам — естественному отбору и генетическому наследованию.

Суть *естественного отбора* заключается в том, что наиболее приспособленные особи лучше выживают и приносят больше потомства, чем менее приспособленные.

Механизм *генетического наследования* состоит в следующем. Почти в каждой клетке любого живого организма имеется набор хромосом, несущих информацию об этом организме. Основная часть хромосомы — цепь ДНК (молекула дезоксирибонуклеиновой кислоты), которая состоит из четырех видов соединений — нуклеотидов, расположенных в определенной последовательности. Нуклеотиды обозначаются буквами A, T, C и G, их порядок следования кодирует все генетические свойства организма.

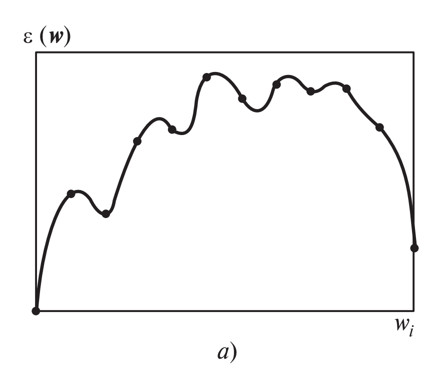
Ген — это отрезок цепи ДНК, отвечающий за определенное свойство особи, например за цвет глаз, тип волос, цвет кожи и т. д. Установлено, что вся совокупность генетических признаков человека кодируется с помощью примерно 60 тыс. генов, суммарная длина которых составляет более 90 млн нуклеотидов.

При размножении живых организмов происходит слияние двух родительских половых клеток: хромосомы родителей сближаются вплотную, затем их цепи ДНК разрываются в нескольких случайных местах и обмениваются своими частями. Таким образом, молекулы ДНК потомков случайным образом приобретают гены как отца, так и матери.

При наследовании возможны мутации — изменения генов в половых клетках одного из родителей. Измененные гены передаются потомку и придают ему новые свойства, отличные от свойств родителей. Если эти новые свойства окажутся полезными, т. е. потомок будет более совершенен, более приспособлен к окружающей среде, то в процессе дальнейшего естественного отбора он выживет и создаст новое, более совершенное потомство.

Таким образом, согласно современной генетической теории, источником совершенствования биологических видов являются механизмы естественного отбора, изменчивости и наследственности. Именно они обеспечивают оптимизационный процесс, безраздельно господствующий во всей живой природе и являющийся неиссякаемым источником совершенства мира, источником его красоты и гармонии.

В 1970-х гг. профессор Мичиганского университета (США) Дж. Холланд [97] предпринял попытку алгоритмизации природного оптимизационного процесса, включая механизмы естественного отбора, изменчивости и наследственности. В алгоритмах Холланда в упрощенной форме была сохранена биологическая терминология, а сами алгоритмы получили название *генетические*. Первоначально такие алгоритмы



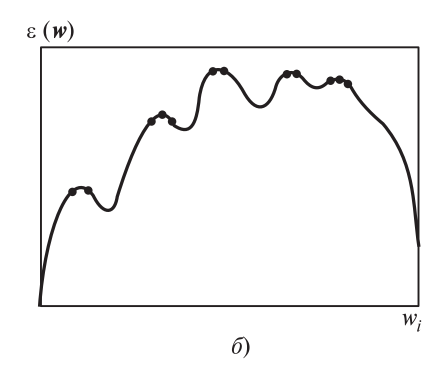


Рис. 6.7. Распределение точек по поверхности целевой функции, соответствующих хромосомам особей начальной популяции (*а*) и особей после смены *n* поколений (*б*)

создавались в познавательных целях, чтобы посмотреть, что получится при моделировании естественной жизни на компьютере, а заодно лишний раз убедиться в справедливости теории Дарвина. Эти модели представляли и представляют интерес для ученых-биологов, которые назвали новое научное направление *Artificial Life*, что в переводе означает *Искусственная жизнь*. Впоследствии выяснилось, что генетические алгоритмы прекрасно справляются с поиском глобальных экстремумов многоэкстремальных функций, в частности — функций ошибок нейронных сетей.

Генетические алгоритмы, предназначенные для обучения нейросетей (оптимизации функции ошибки) работают следующим образом. Сначала создается некоторая начальная *популяция особей*. Каждая особь имеет свою собственную *хромосому* — вектор весовых коэффициентов нейронной сети $\boldsymbol{w} = [w_1, w_2, \ldots w_n]^T$. Для каждой особи вычисляется функция ошибок $\varepsilon(\boldsymbol{w})$ (которую в данном случае называют *функцией фитнеса*, или *целевой функцией*), являющаяся *мерой выживаемости особи,* ее *приспособленности* к окружающей среде и к существованию.

Для удобства дальнейшего изложения вместо задачи минимизации будем рассматривать задачу максимизации, т. е. вместо минимума будем искать максимум. Как известно, эти две задачи легко превращаются одна в другую путем смены знака у целевой функции $\varepsilon(\boldsymbol{w})$.

Первоначальная популяция равномерно распределяется в пространстве оптимизируемых параметров. Таким образом, точки, соответствующие каждой особи, более или менее равномерно распределяются по поверхности целевой функции, как показано на рис. 6.7, *а*.

Отбор особей для *скрещивания*, необходимого для создания *нового поколения*, может основываться на различных принципах. Одним из

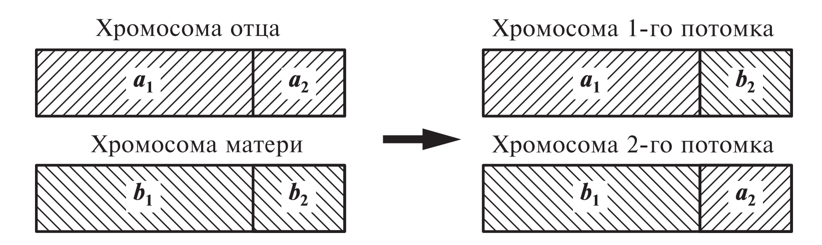


Рис. 6.8. Операция скрещивания, применяемая в генетических алгоритмах

наиболее распространенных считается *принцип элитарности*, согласно которому к скрещиванию допускаются наиболее приспособленные особи, а наихудшие отбраковываются и заменяются вновь создаваемым потомством. Обычно процесс *скрещивания* состоит в том, что *хромосомы родителей* случайным образом рассекаются на две неравные части, после чего они соединяются так, что хромосомы *потомков* содержат часть хромосомы *отца* и часть хромосомы *матери*, как показано на рис. 6.8. В ситуации, изображенной на этом рисунке, после скрещивания хромосомы отца (фрагменты a_1 и a_2) с хромосомой матери (фрагменты b_1 и b_2) образовалась пара новых хромосом, первая из которых имеет фрагменты a_1 и b_2, а вторая — фрагменты b_1 и a_2.

Следующая генетическая операция называется *мутацией* и состоит в замене значения некоторого случайным образом выбранного элемента (гена) случайно выбранного вектора (особи) на новое, случайным образом заданное допустимое значение. Мутации обычно подвергается не более 1...5% бит хромосом всей популяции.

В результате всех этих генетических операций формируется новое поколение, число особей которого обычно равно числу особей предыдущего поколения. Как показывает опыт, новые поколения, созданные в результате отбора, скрещивания и мутаций, в среднем имеют большие значения целевой функции (как показано на рис. 6.7, *б*), т. е. новые поколения являются более совершенными.

Процесс смены поколений завершается после достижения заданного количества итераций ($t \geqslant t_{\max}$, как показано на блок-схеме рис. 6.9) или после того, как одна из особей приобретет заданное максимальное значение целевой функции. Эта особь является победителем, и ее хромосома принимается за окончательное решение оптимизационной задачи, найденное с помощью генетического алгоритма.

В настоящее время известно множество вариантов генетических алгоритмов, различающихся параметрами и способами отбора, скрещивания, мутаций. Практически во все современные нейропакеты генетические алгоритмы включены как эффективные инструменты обучения нейронных сетей. Появились нейропакеты, в которых в функцию фитнеса, помимо синаптических весов, добавлены параметры, харак-

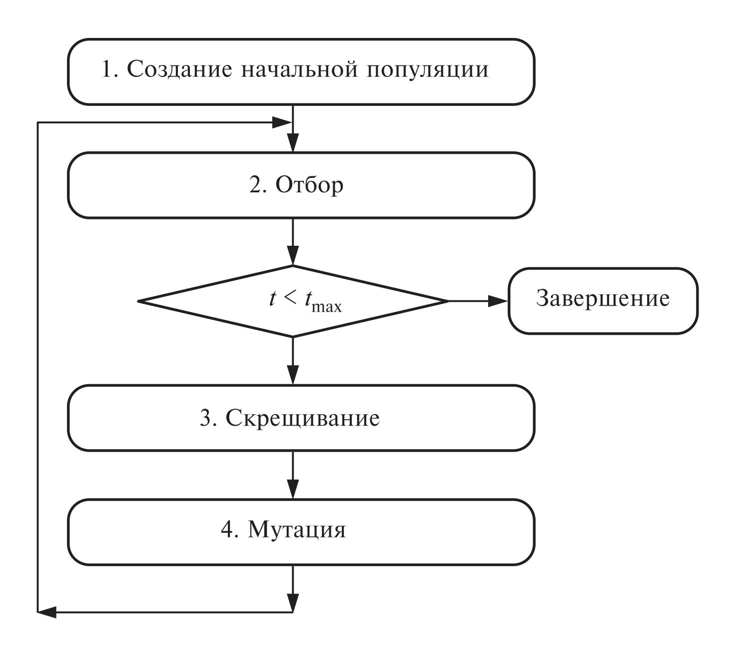


Рис. 6.9. Блок-схема генетического алгоритма

теризующие структуру и тип нейросети. Таким образом, параллельно с обучением нейросети происходит процесс ее оптимизации.

Однако в последнее время генетические алгоритмы нашли независимую от нейросетевых технологий область применения — они лежат в основе многих коммерческих пакетов, широко применяемых для решения разнообразных оптимизационных задач, возникающих в экономике, бизнесе, промышленности и других областях человеческой деятельности.

Вместе с тем многие авторы в качестве недостатков называют сравнительно низкое быстродействие компьютерных программ, реализующих генетические алгоритмы. Эти программы довольно уверенно находят глобальные экстремумы многоэкстремальных функций, однако в ряде случаев, особенно если речь идет об оптимизации одноэкстремальной функции, значительно уступают в скорости градиентным методам, как схематически показано на рис. 6.10.

Попробуем объяснить недостаток генетических алгоритмов, обратившись к природным интерпретациям. Как мы убедились в предыдущем изложении, генетические алгоритмы в упрощенной форме моделируют главные моменты механизма самосовершенствования всего живого, господствующего в дикой природе. Но если вспомнить историю, Чарльза Дарвина многократно критиковали, прежде всего, за чрезмерную упрощенность его гипотез, которые фактически применимы только

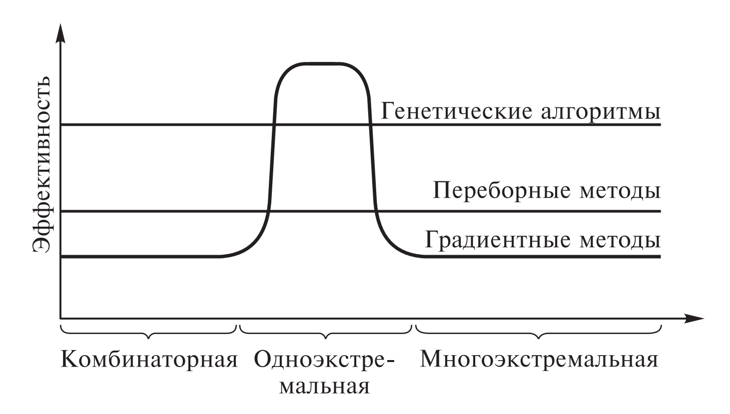


Рис. 6.10. Сравнительная характеристика эффективности оптимизационных методов по [31]

для самых низших форм жизни. Легко заметить, что генетические алгоритмы не учитывают способности высокоразвитых биологических видов улучшать свои качества (совершенствоваться, повышать степень выживаемости — *функцию фитнеса*) в течение жизни. Например, человек занимается спортом, развивая свои физические возможности; получает образование, усиливая умственный потенциал. В том и другом случаях человек повышает значение своей функции фитнеса, не участвуя в скрещиваниях, размножении, мутациях и естественном отборе. Причем такое саморазвитие и самоулучшение может происходить у различных видов как осознанно, так и неосознанно. Например, мускулатуру можно укреплять целенаправленно, занимаясь в спортзале, а можно просто выполнять физическую работу. У животных мускулы укрепляются в процессе движения, когда они ищут пищу, охотятся или убегают от хищников. Таким образом, действие механизма самоулучшения, не учитываемого дарвиновской теорией, можно наблюдать не только у людей. Он характерен для широкого круга достаточно развитых живых организмов. Генетические же алгоритмы (как и само дарвиновское учение) этот дополнительный механизм не учитывают.

Авторами работы [36] была произведена попытка дополнить традиционный генетический алгоритм механизмом *социального самоулучшения особей*, действующем в течение их жизней, а точнее — между рождениями и скрещиваниями. Имитировать механизм социального самоулучшения можно, например, если позволить особям в промежутке между рождениями и скрещиваниями «подниматься» на вершины локальных экстремумов, которые эти особи «видят» и которые им «доступны». Такой подъем можно осуществить, например, путем включения какого-либо градиентного алгоритма оптимизации на шагах генетического процесса, как предложено в [63] и изображено

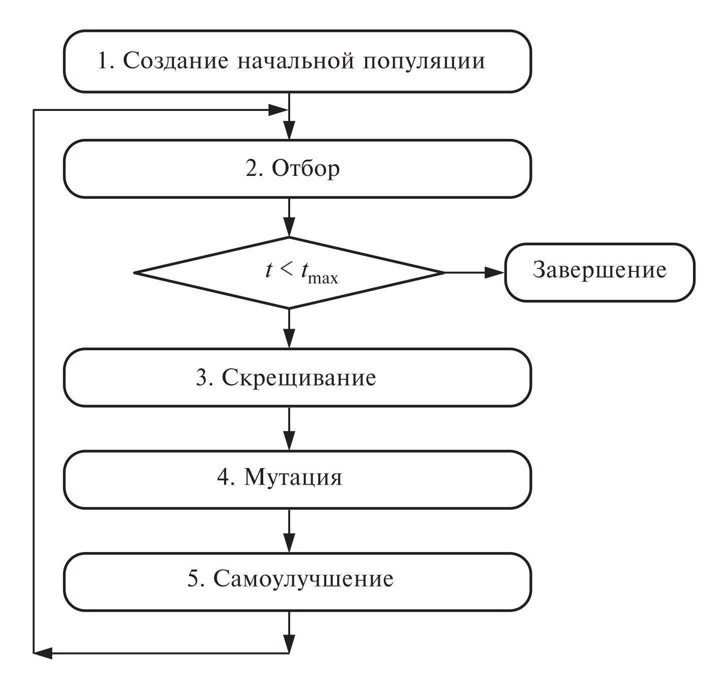


Рис. 6.11. Блок-схема социально-генетического алгоритма: в классическую схему генетического алгоритма (см. рис. 6.9) добавлен блок «5. Самоулучшение»

на блок-схеме рис. 6.11. Как показали вычислительные эксперименты, выполненные в [36], во всех случаях новый *социально-генетический алгоритм* дал более точные результаты за меньшее время.

Возвращаясь к обсуждению Дарвиновской теории, отметим, что в жизни, особенно в молодости, мы всегда стремимся чего-то достичь, подняться на вершины социальной лестницы, которые мы видим и которые нам представляются перспективными. Обычно нас окружает множество таких вершин, но какой из них отдать предпочтение, каждый решает сам. Так, окончив обучение в школе, молодые люди выбирают, что делать дальше: идти работать, создавать свой собственный бизнес, служить в армии, поступать в колледж или в университет. Все это локальные вершины функции фитнеса, на которые каждый взбирается самостоятельно, без помощи Чарльза Дарвина. Кто сделал правильный выбор и добился действительного успеха, чей экстремум оказался выше — решает Жизнь. К моменту таинства отбора и скрещивания Человек, в отличие от животного, подходит отнюдь не в том виде, в каком он родился, а с определенным багажом социальных критериев.

Механизм оптимизации всего живого, действующий, согласно Дарвину, в дикой природе, по нашему мнению, несколько отличается

от того, что происходит в человеческом обществе. Попытка дополнить дарвиновскую эволюцию социальным фактором привела к тому, что новый социально-генетический алгоритм оказался значительно эффективнее традиционного генетического.

На рис. 6.10 были приведены заимствованные из [31] графики эффективности применения оптимизационных методов для решения трех видов задач: комбинаторных, одноэкстремальных и многоэкстремальных. Как видно из рисунка, генетические алгоритмы наиболее эффективны при решении комбинаторных и многоэкстремальных задач оптимизации; если же целевая функция имеет только один экстремум, то генетические алгоритмы уступают градиентным.

Как показано в [36], предложенный в этой работе *социально-генетический* метод превосходит по эффективности традиционный генетический алгоритм как в случае одноэкстремальных, так и в случаях многоэкстремальных целевых функций. В случае одноэкстремальной функции социально-генетический метод практически вырождается в градиентный, так как всю работу поиска экстремума выполняет его градиентная составляющая. Поэтому эффективности их применения на одноэкстремальных задачах практически совпадают. На основании сказанного, примерную эффективность нового метода на графиках рис. 6.10 можно представить пунктирной линией, как изображено на рис. 6.12.

В заключение отметим, что социально-генетический алгоритм довольно хорошо показал себя при обучении и оптимизации нейронных сетей и его можно рекомендовать разработчикам современных нейропакетов.

Отметим также, что приведенная здесь попытка учета социального фактора [36, 63] является еще одним шагом на пути совершенствования искусственно-интеллектуальных парадигм путем более полного моделирования естественных механизмов, причем не только биологических и природных, но и социальных.

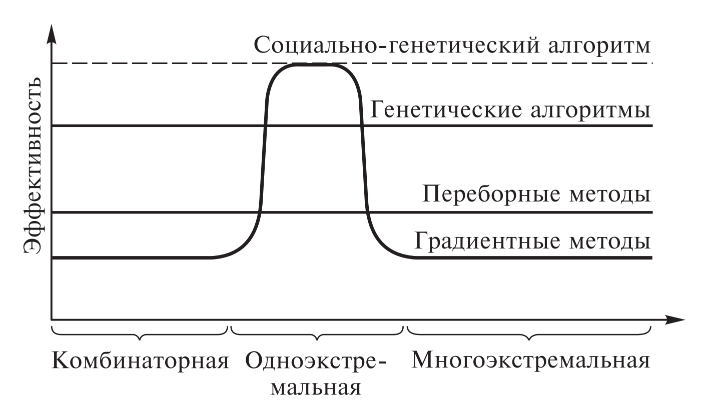


Рис. 6.12. Сравнительная характеристика эффективности оптимизационных методов по [36]

6.5. ДОПОЛНИТЕЛЬНЫЕ РЕКОМЕНДАЦИИ ПО ПРОЕКТИРОВАНИЮ И ОБУЧЕНИЮ ПЕРСЕПТРОНОВ

Многие специалисты, занимающиеся практическим применением нейросетевых технологий, склонны называть проектирование и обучение нейронных сетей скорее искусством, чем наукой. Действительно, несмотря на фундаментальный характер рассмотренных в этой главе теорем, несмотря на рекордные количества, строгость и красоту их доказательств, программисты в своей практической работе часто руководствуются так называемыми эвристическими правилами, рекомендациями своих коллег, собственной интуицией и только им одним известными приемами — таинственными know-how. Приведем некоторые из них.

Выбор входных параметров нейросетевой модели. От удачного выбора входных параметров нейросетевой модели во многом зависит успех ее создания. Прежде всего, необходимо понимать, что не все параметры предметной области влияют на выходной вектор $\boldsymbol{Y}$. Например, знание температуры тела вряд ли поможет в постановке диагноза больного, если задача состоит в выявлении у него только психических отклонений.

Те параметры, которые не оказывают влияния на вектор $\boldsymbol{Y}$, называют *незначимыми* для этого выходного вектора. Естественно, что незначимые параметры не следует включать в список параметров входного вектора $\boldsymbol{X}$.

Однако на практике часто бывает трудно и даже невозможно установить, какие из параметров предметной области являются значимыми, а какие нет. Поэтому на первом этапе мы рекомендуем включать в вектор $\boldsymbol{X}$ как можно больше параметров, исключая только те, незначимость которых представляется очевидной.

После первоначального создания и обучения нейронной сети незначимые параметры могут быть выявлены следующими тремя способами.

1. Анализ значений весовых коэффициентов входных нейронов. Если окажется, что у какого-либо входного нейрона синаптические веса значительно меньше, чем у других нейронов обученной сети, то этот входной нейрон, скорее всего, соответствует незначимому параметру вектора $\boldsymbol{X}$.

2. Возмущение значений входных параметров и анализ реакции сети на эти возмущения. Если обученная сеть не реагирует или слабо реагирует на изменения значения какого-либо входного параметра, то этот параметр не является значимым.

3. Поочередное исключение входных нейронов и наблюдение за ошибкой обобщения сети. Если после исключения какого-либо входного нейрона ошибка обобщения ε_T увеличилась, то входной параметр, соответствующий этому входному нейрону, является значимым.

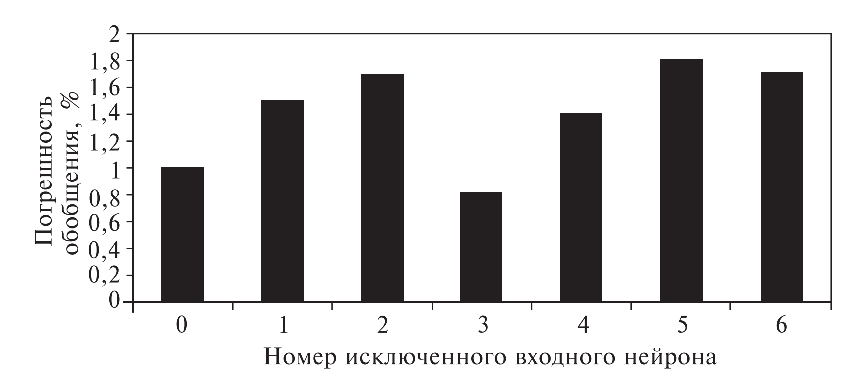


Рис. 6.13. Пример изменения погрешности нейронной сети при поочередном исключении входных нейронов

В противном случае входной параметр не является значимым и соответствующий ему входной нейрон можно исключить.

Процесс выявления незначимых входных параметров рекомендуется сопровождать графической иллюстрацией — гистограммой (рис. 6.13), на которой нулевой столбец показывает погрешность обобщения сети при сохранении всех входных нейронов, первый столбец — погрешность обобщения сети при исключенном первом входном нейроне; второй столбец — погрешность при исключенном втором входном нейроне и т. д. На рис. 6.13 третий нейрон сети не является значимым, поэтому его рекомендуется исключить.

4. Иногда имеет смысл вместо поочередного исключения входных нейронов поочередно подавать на них произвольные сигналы датчиком случайных чисел. Автор этого метода [56] обосновывает его тем, что исключение входных нейронов соответствует случаю подачи на них нулевого сигнала. Но нулевой входной сигнал может быть обусловлен условиями конкретной задачи и иметь физический смысл, что, по мнению авторов, может привести к ошибочным оценкам значимости.

После выявления и исключения входных нейронов, соответствующих незначимым параметрам, качество нейросети, как правило, несколько улучшается, так как снижается ее размерность. Однако надо понимать, что при слишком малом количестве входных параметров нейросети может не хватить данных для выявления закономерностей предметной области. В частности, могут появиться так называемые конфликтные примеры, имеющие одинаковые входные, но различные выходные параметры.

Выявление конфликтных примеров. *Конфликтными*, или *противоречивыми*, будем называть примеры, имеющие одинаковые входные векторы и различающиеся выходные векторы. Например, конфликтными являются примеры: $2 \cdot 2 = 4$ и $2 \cdot 2 = 5$. После обучения на таких примерах сеть усвоит: $2 \cdot 2 = 4,5$. Погрешность обучения при этом со-

ставит $\varepsilon_L = \frac{4,5-4}{4} \times 100\% = 12,5\%$ и никогда не станет ниже этой величины, какие бы методы обучения мы ни применяли.

Наличие конфликтных примеров в обучающем множестве обычно является результатом закравшейся ошибки (как в нашем случае) либо некорректной постановки задачи. Например, при построении системы медицинской диагностики входному параметру «Повышенная температура тела» могут соответствовать несколько диагнозов: «Ангина», «ОРЗ», «Пневмония». Это значит, что при постановке задачи, помимо температуры, надо учитывать еще и другие значащие входные параметры, такие как «Кашель», «Слабость», «Хрипы в легких», «Затемнения на рентгеновских снимках» и т. д. Добавление значащих входных параметров (и, соответственно, входных нейронов) позволяет избежать наличия конфликтных примеров в обучающем множестве и сделать содержательную постановку задачи корректной.

Обнаружить конфликтные примеры можно путем визуального анализа множества примеров предметной области, сервисными средствами программ-редакторов либо путем запуска специального алгоритма поиска конфликтных примеров, имеющегося в некоторых нейропакетах.

Определение необходимого количества примеров. Множество примеров поведения предметной области может формироваться из различных источников. Это может быть информация из сети Интернет, из архивных материалов предприятий, из результатов социологических опросов и анкетирования и пр. Обычно считается, что чем больше примеров, тем лучше. Однако излишне большое количество примеров приводит к неоправданным затратам машинного времени на обучение сети. Автором работы [19] для определения объема обучающего множества рекомендуется эмпирическая формула:

$$Q = 7 \cdot N_x + 15, \tag{6.4}$$

где N_x — количество входных параметров нейросетевой модели; Q — количество примеров обучающего множества. Однако, как показывает наш эмпирический опыт, эта формула более приемлема при построении регрессионных математических моделей. Для нейронных же сетей достаточное количество примеров зачастую в 2–4 раза меньше, чем определенное по формуле (6.4).

В практике нейросетевого моделирования нередки случаи, когда не удается собрать достаточное количество примеров поведения предметной области, и условие формулы (6.4) не выполняется, поэтому возникает необходимость более тщательного тестирования нейросетевой модели. В этом случае рекомендуется прием, называемый *многократной перекрестной проверкой* (multifold cross-validation). Суть этого приема заключается в следующем.

Множество примеров предметной области, как обычно, разбивается на обучающее L_1 и тестирующее T_1 в отношении 10 : 1. Нейронная

сеть обучается на примерах множества L_1, а на примерах множества T_1 вычисляется ошибка тестирования ε_{T_1}. Затем множества L_1 и T_1 объединяются. Объединенное множество вновь разбивается на обучающее L_2 и тестирующее T_2 в том же соотношении (10 : 1) таким образом, чтобы ранее входившие в T_1 примеры в множестве T_2 отсутствовали. Нейросеть обучается на множестве L_2, а на множестве T_2 вычисляется ошибка тестирования ε_{T_2}. И так далее, всего до 10 раз. Если максимальная из $\varepsilon_{T_1}, \varepsilon_{T_2}, \ldots, \varepsilon_{T_{10}}$ погрешность оказывается в пределах приемлемого значения, то любая из полученных нейросетей считается пригодной для последующего использования.

Порядок представления обучающих примеров. При обучении сети методом обратного распространения ошибки или другими методами, рассмотренными в § 6.3, обучающие примеры рекомендуется представлять не в порядке их следования от первого до последнего, а выбирать случайным образом. По мнению некоторых специалистов [93], это придает алгоритму обучения более стохастический характер и в некоторых случаях позволяет избегать попадания в локальные минимумы.

Выявление выбросов. Под *выбросами* понимаются примеры поведения предметной области, которые в силу каких-либо случайных причин, не характерных для рассматриваемой предметной области, значительно отличаются по своим параметрам от другой статистической информации. Их появление может быть связано с ошибками, допущенными при сборе информации (например, забыли поставить ноль, в результате чего ошиблись на порядок) либо другими случайными причинами (погрешностями измеряющих приборов, сбоев в работе оборудования, форс-мажорными обстоятельствами и т. д.), не отражающими закономерностей предметной области.

Присутствие выбросов в статистических данных является весьма распространенным явлением, отрицательно влияющим на качество их последующей обработки и использования. Если в простейших случаях выбросы обнаруживаются сравнительно легко и даже видны невооруженным глазом, то в более сложных многомерных зависимостях при больших объемах информации выявление и удаление выбросов представляет собой непростую математическую проблему.

В работе [57] для обнаружения выбросов в статистических данных предлагается использовать персептрон, в частности его свойство плохо обучаться на примерах, имеющих выбросы, не подчиняющиеся закономерностям предметных областей. Идея алгоритма состоит в поочередном исключении примеров из обучающего множества и наблюдении за погрешностью нейросети, обученной на этих урезанных обучающих множествах. Если пример обучающего множества является выбросом и выпадает из закономерности, характерной для предметной области, то его удаление из обучающего множества приводит к умень-

шению погрешности обучения сети и повышению ее обобщающих свойств. Исключение обычных примеров существенного влияния на качество сети, как правило, не оказывает.

Эту идею можно реализовать несколькими способами. Приведем описание одного из них, названного в [57] *анализом ошибки обучения*. Этот метод основывается на том факте, что при удалении выброса из обучающего множества нейронной сети будет легче выявить закономерности предметной области, сеть быстрее и лучше обучится. При этом уменьшится среднеквадратичная ошибка обучения, вычисляемая после завершения процесса обучения по формуле

$$\varepsilon_L = \frac{\sqrt{\frac{1}{JQ}\sum\limits_{q=1}^{Q}\sum\limits_{j=1}^{J}(y_{qj} - d_{qj})^2}}{M[d_{qj}]} \times 100\%, \tag{6.5}$$

где y_{qj} — значение j-го выхода нейросети для q-го обучающего примера; d_{qj} — желаемое значение j-го выхода для q-го обучающего примера; J — число нейронов в выходном слое; Q — количество примеров в обучающем множестве. $M[d_{qj}]$ — математическое ожидание всех желаемых выходов на всех обучающих примерах.

В качестве демонстрационного примера возьмем обучающее множество из девяти компонент, помещенных в табл. 6.1. Заметим, что третий пример в этом множестве выпадает из общей линейной закономерности и, следовательно, его можно рассматривать как выброс.

На рис. 6.14 приведена гистограмма, полученная в результате работы предлагаемого алгоритма. Здесь i-й столбец ($i = 1, 2, \ldots, 9$) изображает значение среднеквадратичной ошибки (6.5) при обучении сети на обучающем множестве с исключенным i-м примером. Как видно из рисунка, при исключении из обучающего множества третьего примера, являющегося выбросом, среднеквадратичная ошибка значительно уменьшается.

Для сравнения на рис. 6.15 приведена аналогичная гистограмма для обучающих примеров той же предметной области (см. табл. 6.1), но не имеющих выбросов, поскольку $d_3 = 3{,}0$. Среднеквадратичные ошибки обучения сети здесь распределились более равномерно.

Отметим, что вид гистограммы на рис. 6.14 зависит от структуры нейронной сети. В данном случае использован персептрон с двумя сигмоидными нейронами на единственном скрытом слое. При увеличении числа скрытых нейронов разница между высотой столбцов гистограммы уменьшается, вследствие этого «плохие» примеры становятся не столь заметными. Поэтому для обнаружения посторонних выбросов рекомендуется использовать нейросети с минимальным количеством нейронов.

Выбор активационных функций. Многие авторы рекомендуют отдавать предпочтение активационным функциям, обладаю-

Таблица 6.1

Обучающее множество с выбросом в третьем примере

№ примера	x	d
1	1,0	1,0
2	2,0	2,0
3	3,0	**3,5**
4	4,0	4,0
5	5,0	5,0
6	6,0	6,0
7	7,0	7,0
8	8,0	8,0
9	9,0	9,0

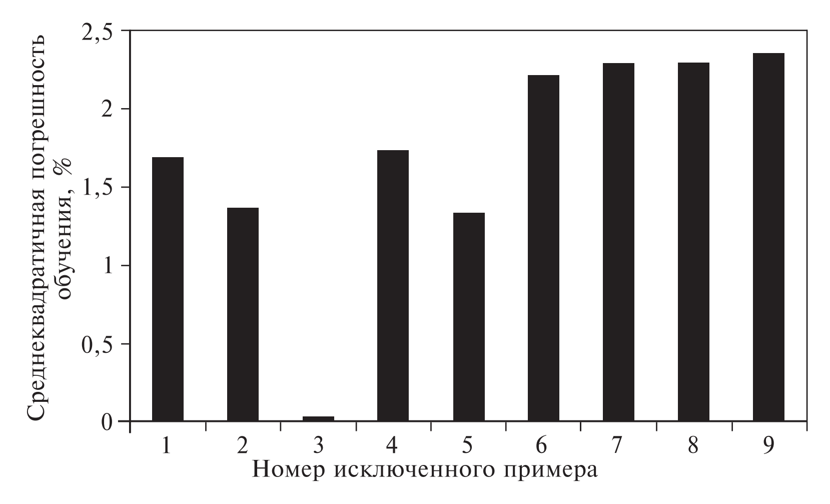


Рис. 6.14. Распределение среднеквадратичной ошибки обучения ε_L при поочередном исключении примеров обучающего множества, представленного в табл. 6.1

щим свойством антисимметрии, для которых выполняется равенство $y(-S) = -y(S)$. Таким свойством обладают, например, функции: $y = \frac{\alpha S}{1 + |\alpha S|}$ и $y = \operatorname{th} \alpha S$, которые также называют сигмоидными.

В работе [54] рекомендуется функция, предложенная *Ле-Каном*:

$$y = a \operatorname{th}(bS), \tag{6.6}$$

где $a = 1{,}7159$; $b = 2/3$. При таких параметрах функция гиперболического тангенса также имеет вид сигмоиды, но отличается рядом полезных свойств:

1) $y(1) = 1$ и $y(-1) = -1$;

2) в начале координат первая производная близка к единице;

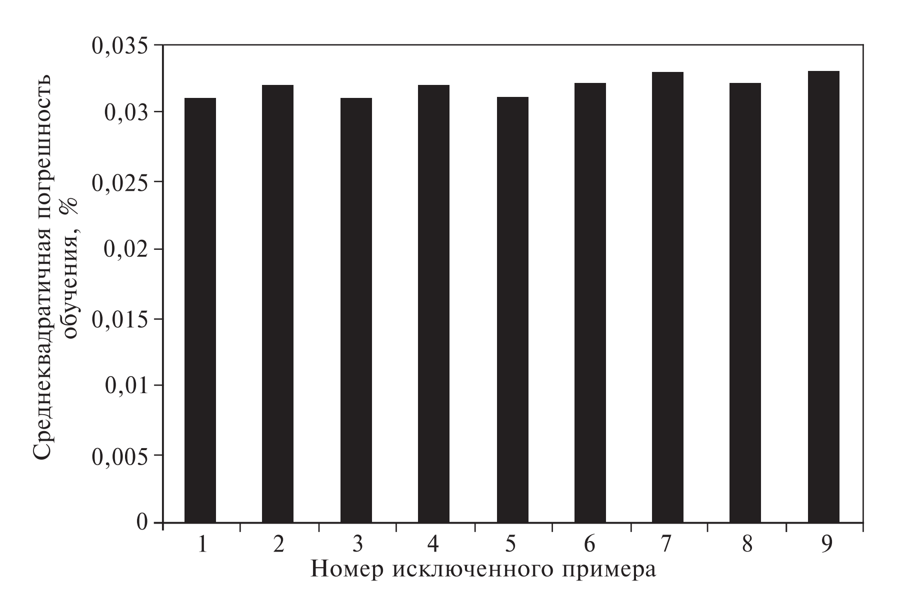


Рис. 6.15. Пример распределения среднеквадратичной ошибки обучения ε_L при поочередном исключении примеров обучающего множества, не имеющего выбросов

3) вторая производная достигает своего максимального значения при $S = 1$.

Авторы работы [92] предлагают логарифмическую активационную функцию $y = \ln(S + \sqrt{S^2 + 1})$. Эта функция, в отличие от функции Ле-Кана и других сигмоидных функций, имеет неограниченную область изменения, что позволяет избежать нежелательного эффекта, называемого *параличом сети*. Как уже отмечалось, явление паралича сети состоит в потере чувствительности сети к вариациям весовых коэффициентов. Это сопровождается замиранием процесса обучения при попадании взвешенных сумм входных сигналов нейрона в область насыщения сигмоиды.

Инициализация синаптических весов. Согласно методике обучения персептрона, начальные значения сил синаптических связей задаются датчиком случайных чисел. По мнению авторов работы [92], во избежание наступления паралича сети желательно, чтобы при этих начальных значениях нейроны сети индуцировали значения, лежащие где-то в переходе между линейной частью сигмоидной функции активации и областью насыщения. Согласно расчетам [92], для функции активации Ле-Кана эта цель будет достигнута, если равномерное распределение, из которого выбираются начальные значения синаптических весов, задавать специальным образом. А именно, распределение

должно иметь нулевое среднее значение и дисперсию, обратную квадратному корню из количества синаптических связей нейрона.

Предобработка обучающих примеров и интерпретация ответов. Параметры, описывающие предметную область, могут иметь самый разнообразный характер. Это могут быть числа с различными диапазонами изменений, могут быть качественные характеристики, такие как цвет волос и глаз пациента, могут быть даты, содержащие число, месяц, год, могут быть графические объекты. Поскольку нейронная сеть в состоянии обрабатывать только числа, то вся нечисловая информация должна быть закодирована в числовом виде. Числовую информацию, приготовленную для нейросетевой обработки, желательно нормализовать, т. е. выровнять диапазоны изменения величин, например ограничив их интервалом $[-1, 1]$. Сделать это можно с помощью простейшего линейного преобразования:

$$\tilde{x}_n = \frac{x_n - x_{n\,\min}}{x_{n\,\max} - x_{n\,\min}}(B - A) + A, \tag{6.7}$$

где x_n и $\tilde{x}_n$ — значения исходного и нормализованного n-го параметра предметной области, подаваемого на n-й входной нейрон нейросети; $[x_{n\,\min}, x_{n\,\max}]$ — реальный диапазон изменения n-го параметра; $[A, B]$ — приемлемый диапазон изменения входных сигналов, например $A = -1$, $B = 1$.

Желаемые выходные сигналы персептрона также должны быть закодированы в приемлемой форме и нормализованы в приемлемом диапазоне $[a, b]$. Это значит, что при формировании обучающего вектора $\boldsymbol{D}$ следует применить формулу нормализации, аналогичную (6.7):

$$\tilde{d}_m = \frac{d_m - d_{m\,\min}}{d_{m\,\max} - d_{m\,\min}}(B - A) + A, \tag{6.8}$$

где d_m и $\tilde{d}_m$ — заданное и нормализованное значение m-й компоненты вектора $\boldsymbol{D}$. Очень важно, чтобы желаемые выходные сигналы после нормализации укладывались с некоторым запасом ε в область значений сигмоидной функции активации. В противном случае может возникнуть паралич сети — алгоритм обучения устремит синаптические веса к бесконечности и не сможет обеспечить снижение ошибки обучения. Во избежание этого эффекта в [92] для случая использования активационной функции Ле-Кана (6.6) рекомендуется задавать $A = -a + \frac{\varepsilon}{2}$, $B = a - \frac{\varepsilon}{2}$, где $\frac{\varepsilon}{2} = 0,7159$.

Обученный на таком обучающем множестве персептрон будет формировать выходной вектор $\tilde{\boldsymbol{Y}}$, содержащий значения, приведенные к диапазону $[A, B]$. Поэтому к ним должно быть применено преобразование, обратное преобразованию (6.8):

$$y_m = \frac{\tilde{y}_m - A}{B - A}(d_{m\,\max} - d_{m\,\min}) + d_{m\,\min}. \tag{6.9}$$

Таким образом, персептрон можно применять для моделирования предметной области, описываемой числовыми параметрами любого

диапазона. При этом входные и выходные параметры желательно нормализовать — преобразовать к приемлемым диапазонам. Естественно, что ответы персептрона следует денормализовать путем применения обратного преобразования.

Рассмотрим возможности интерпретации ответов персептрона при решении задач классификации. Примеры решения задач классификации приводились ранее. Это задачи постановки диагнозов сложных технических устройств (см. § 5.3) и диагнозов заболеваний человека (см. § 5.2). В этих случаях персептроны строились таким образом, что каждому классу (каждому диагнозу) отводился свой выходной нейрон. Размерность обучающих векторов $\boldsymbol{D}$ совпадала с количеством выходных нейронов, а его компонентам d_m задавалось значение 1, если для подготовленного вектора $\boldsymbol{X}$ имел место диагноз, за который «отвечает» m-й выходной нейрон, и значение 0, если диагноз был другим. При появлении на входе персептрона нового вектора параметров, не встречавшихся в обучающем множестве, персептрон вычислял вектор $\boldsymbol{Y}$, который нужно интерпретировать с целью получения заключения о классификации объекта. Наиболее распространенный способ интерпретации состоит в том, что выходному сигналу m-го нейрона присваивается значение 1, если $y_m \geqslant 0{,}5$, и 0 — в противном случае. Причем логично полагать, что чем выше значение y_m, тем более вероятна правильность постановки диагноза m-й болезни или диагноза неисправности. И наоборот, чем меньше значение y_m, тем вероятнее отсутствие m-й болезни или неисправности. В этом случае говорят, что y_m можно расценивать как *функцию принадлежности* объекта какому-либо классу или как *меру уверенности* ответа персептрона.

Таким образом, мы видим, что персептрон дает нечеткий ответ, оценивая вероятность возможной ошибки. Подобно добросовестному врачу-диагносту, персептрон может ответить, что у больного наверняка есть инфаркт миокарда, однако с вероятностью 65% он подозревает, что у больного был порок сердца, и на 90% персептрон уверен, что ишемической болезни сердца у больного нет.

Приведем пример интерпретации ответов нейронной сети, решающей задачу детекции лжи, если эта сеть имеет один выходной нейрон, а в множестве обучающих примеров правдивый ответ клиента кодировался единицей, а ложный — нулем. Тогда выходной сигнал сети $y = 0{,}9$ можно интерпретировать 90%-й уверенностью сети в том, что клиент сказал правду. В случае $y = 0,75$ степень уверенности сети в правдивом ответе составляет 75%, а случай $y = 1,2$ можно интерпретировать как абсолютную уверенность (120%) сети в том, что ответ клиента является правдой. Случай $y = 0{,}4$ интерпретируется 60%-й уверенностью сети в том, что ответ клиента является ложью. Случай $y = 0{,}1$ соответствует 90%-й уверенности в ложном ответе клиента,

а случай $y = -0{,}1$ означает абсолютную (на 110%) уверенность сети в ложности ответа клиента.

В заключение отметим, что, помимо рассмотренных способов подготовки данных и интерпретации ответов нейросетей, существует множество других приемов [13–15, 41], преследующих аналогичные цели.

Добавление шума в обучающие примеры. Некоторые авторы [41] полагают, что улучшить обобщающие свойства сети можно не только за счет оптимизации ее архитектуры, но и путем специальной подготовки множества обучающих примеров. В частности, путем добавления шума в обучающие примеры можно добиться снижения чувствительности сети к вариациям входных сигналов, если эти вариации находятся в определенных допустимых пределах.

Вспомним, что процесс обучения персептрона выполняется на множестве обучающих примеров — пар входных и выходных векторов: $\boldsymbol{X}_q$–$\boldsymbol{D}_q$, $q = 1, 2, \ldots, Q$. Напомним, что q — номер обучающего примера, а Q — их общее количество.

При обучении сети минимизируется квадратичная невязка между желаемыми выходными векторами $\boldsymbol{D}_q$ и теми значениями этих векторов, которые образуются на выходе персептрона в действительности $\boldsymbol{Y}_q$, т. е. минимизируется целевая функция

$$\varepsilon = \frac{1}{2}\sum_{q=1}^{Q} \|\boldsymbol{D}_q - \boldsymbol{Y}_q\|^2. \qquad (6.10)$$

Персептрон выполняет многомерное отображение входного вектора $\boldsymbol{X}_q$ на вектор $\boldsymbol{Y}_q$, т. е. $\boldsymbol{Y}_q = \boldsymbol{f}(\boldsymbol{X}_q)$, где $\mathbf{f}$ — матрица, составленная из функций активации нейронов. С учетом этого равенство (6.10) перепишем в виде:

$$\varepsilon = \frac{1}{2}\sum_{q=1}^{Q} \|\boldsymbol{D}_q - \boldsymbol{f}(\boldsymbol{X}_q)\|^2. \qquad (6.11)$$

Добавление шума в обучающие примеры предполагает модификацию входного вектора — добавление к нему некоторого случайного вектора $\boldsymbol{\delta}_q$, математическое ожидание которого равно нулю. Таким образом, вместо целевой функции (6.11) минимизируется целевая функция

$$\varepsilon = \frac{1}{2}\sum_{q=1}^{Q} \|\boldsymbol{D}_q - \boldsymbol{f}(\boldsymbol{X}_q + \delta_q)\|^2. \qquad (6.12)$$

В результате сеть тренируется не только на примерах обучающего множества, но и на таких примерах, в которых входной вектор имеет незначительные случайные искажения. Поэтому обобщающие свойства сети улучшаются.

В заключение отметим, что добавление шума в обучающее множество, как правило, дает положительные результаты в случае больших

(более 300 элементов) обучающих множеств, причем дисперсия случайных параметров вектора шума δ_q должна подбираться индивидуально для каждой конкретной задачи.

Декомпозиция сети по числу выходных нейронов. При разработке нейросетевых математических моделей с несколькими выходными параметрами иногда бывает полезно применять прием, заключающийся в том, что вместо классической схемы нейронной сети персептронного типа с N входами и M выходами (рис. 6.16, *слева*) использовать M нейросетей с N входами и одним выходом (рис. 6.16, *справа*), объединенных общим интерфейсом. Структуру каждой такой сети, естественно, следует оптимизировать отдельно из условия обеспечения минимальности ошибки тестирования каждой сети.

Вычислительные эксперименты показали, что применение указанного приема, как правило, позволяет снижать общую погрешность нейросетевых моделей на 3–6%.

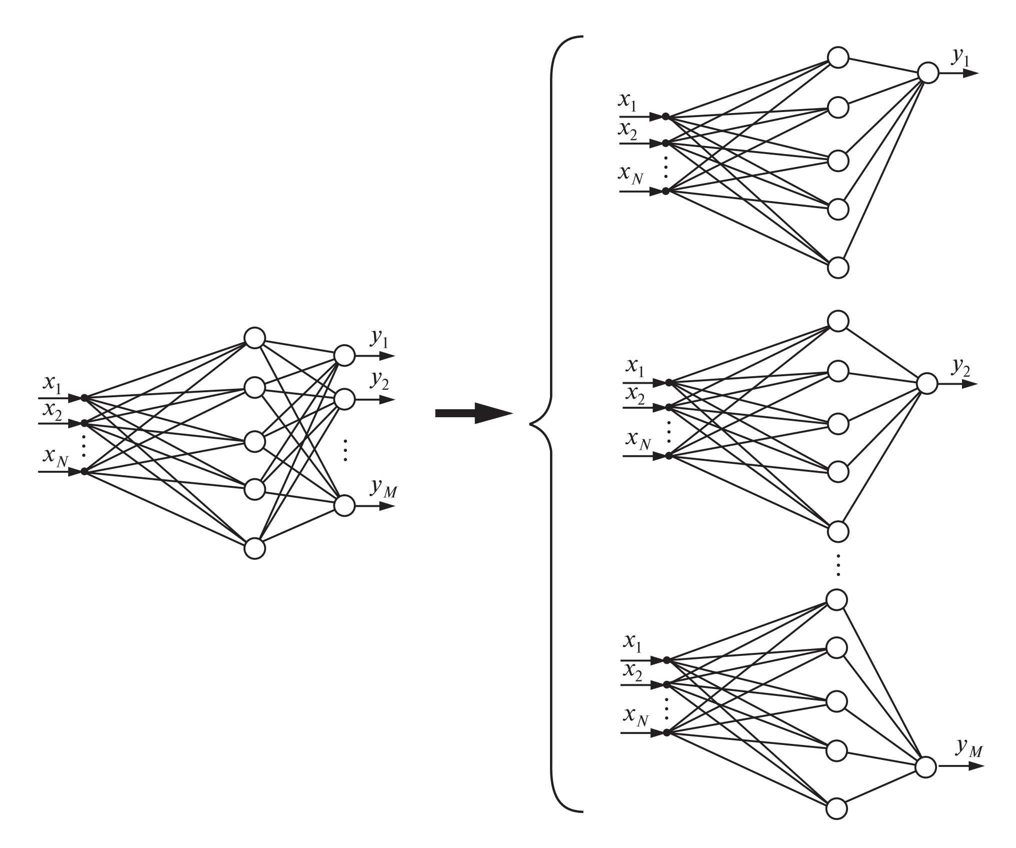


Рис. 6.16. Вместо классической схемы персептрона с N входами и M выходами применяются M персептронов с N входами и одним выходом, объединенных общим интерфейсом

6.6. АЛГОРИТМ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ МЕТОДОМ НЕЙРОСЕТЕВОГО МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

Завершая изучение метода математического моделирования на основе персептронов, в качестве итога приведем примерный алгоритм его применения (рис. 6.17).

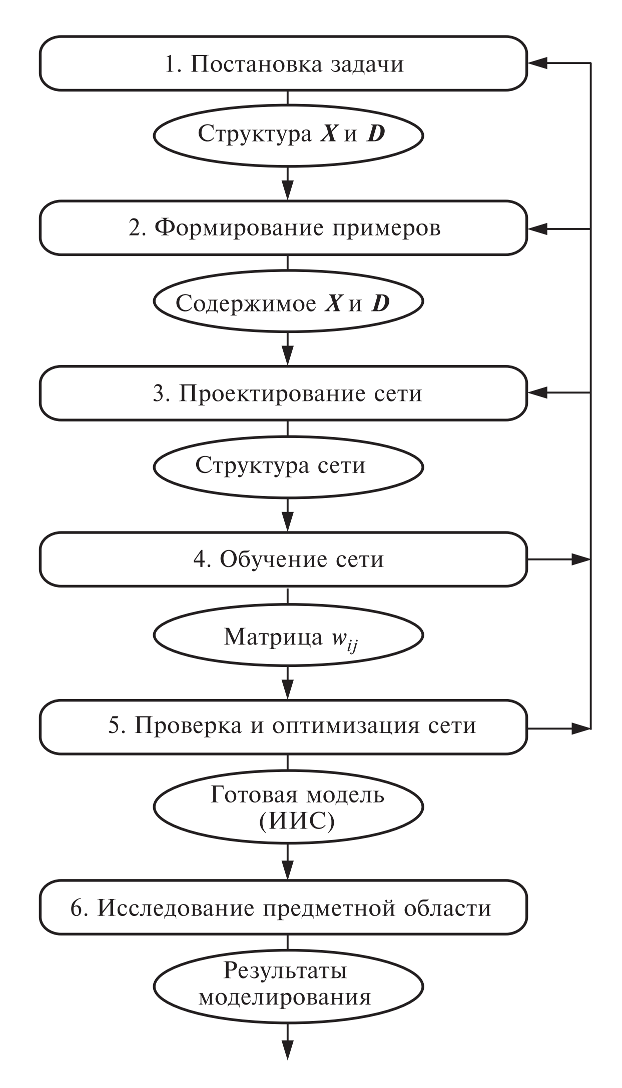


Рис. 6.17. Рекомендуемая схема применения метода нейросетевого математического моделирования

Этап 1. Постановка задачи

На этом этапе определяются цели моделирования, устанавливаются входные и выходные параметры модели, а также структура (состав и длина) входного вектора $\boldsymbol{X}$, и выходного вектора $\boldsymbol{D}$.

В качестве компонент входного вектора $\boldsymbol{X}$ важно выбрать *значимые* параметры — те, которые оказывают существенное влияние на результат. Если же имеются сомнения в значимости того или иного входного параметра, то его лучше включить во входной вектор, рассчитывая, что в последующем с помощью создаваемой нейросети можно будет оценить степень влияния этого параметра на результат. Если его влияние окажется слабым, то в последующей работе исключить этот параметр.

Выходной вектор $\boldsymbol{D}$ формируется таким, чтобы его компоненты давали возможность получить ответы на все поставленные вопросы.

Компоненты входного вектора $\boldsymbol{X}$ и выходного вектора $\boldsymbol{D}$ представляют собой числа. Это могут быть значения каких-либо величин, например температуры тела, артериального давления, частоты пульса и др. Это могут быть также числа, кодирующие наличие или отсутствие каких-либо признаков, например единица, если пол мужской, и двойка, если пол женский. В некоторых случаях, если данные нечеткие и есть сомнение в их правильности, полезно кодировать оценку их вероятности. Например, если у врача есть сомнения в правильности выставляемого диагноза, то он может закодировать не сам диагноз, а его вероятность или степень развития болезни, применяя десяти- или стобалльную систему оценки.

Этап 2. Формирование примеров

На этом этапе формируется содержимое входных и выходных векторов. В результате создается множество пар $\boldsymbol{X}_q$–$\boldsymbol{D}_q$ ($q = 1, \ldots, Q$), где такая пара составляет пример, характеризующий предметную область. Значения компонент векторов $\boldsymbol{X}_q$ и $\boldsymbol{D}_q$ могут быть сформированы различными способами: получены путем проведения социологических опросов, экспертных оценок, анкетирования, специальных экспериментов над предметной областью, взяты из средств массовой информации, из архивных материалов организаций, социальных сетей, интернет-форумов и из других источников.

Числовую информацию, приготовленную для ввода в нейросеть, желательно нормировать — привести к диапазону $[0, 1]$ или $[-1, 1]$.

Все множество примеров разбивают на обучающее $\boldsymbol{L}$ и тестирующее $\boldsymbol{T}$ (см. рис. 6.1). Обычно объем тестирующего множества выбирают не менее 10–15% от обучающего. Необходимый минимальный объем обучающего множества зависит от задачи. Ранее для расчета минимально допустимого объема обучающего множества рекомендовалась

формула: $Q = 7 \cdot N_x + 15$ [19], в которой N_x — количество входных параметров нейросетевой модели. Однако в практике применения нейросетевых технологий встречаются случаи, когда для решения задач хватает значительно меньшего количества обучающих примеров.

Помимо обучающего множества $\boldsymbol{L}$ и тестирующего множества $\boldsymbol{T}$, формируется еще и подтверждающее множество $\boldsymbol{P}$ из примеров, принадлежащих той же самой предметной области, но не пересекающееся ни с множеством $\boldsymbol{L}$, ни с множеством $\boldsymbol{T}$ (см. рис. 6.1). Объем множества $\boldsymbol{P}$ обычно рекомендуется не более 10% от объема множества $\boldsymbol{L}$.

Этап 3. Первоначальное проектирование сети

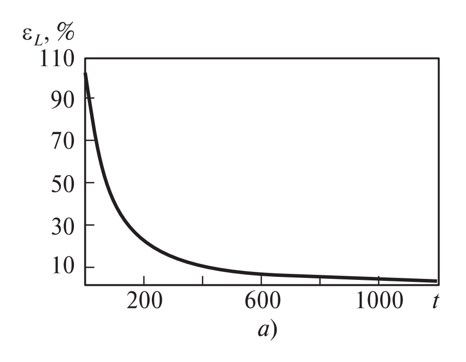
Структура персептрона выбирается из следующих соображений:

1) количество входных нейронов N_x должно быть равно размерности входного вектора $\boldsymbol{X}$;
2) количество выходных нейронов N_y должно быть равно размерности выходного вектора $\boldsymbol{D}$;
3) количество скрытых слоев, согласно теореме Арнольда — Колмогорова — Хехт-Нильсена (см. § 6.1), должно быть не менее одного; на последующих этапах количество скрытых слоев может корректироваться, если это позволит улучшить качество работы сети;
4) количество нейронов в скрытых слоях первоначально рассчитывается с помощью формул (6.1), (6.2); на последующих этапах количество нейронов в скрытых слоях также может корректироваться, если это позволит улучшить качество работы сети;
5) активационные функции скрытых нейронов, согласно теореме Арнольда — Колмогорова — Хехт-Нильсена, рекомендуется задать сигмоидными, однако в дальнейшем их вид может быть изменен, если это позволит улучшить качество работы сети.

При корректировке структуры персептрона следует иметь в виду, что увеличение количества скрытых нейронов обычно позволяет добиться меньшей ошибки обучения. Однако чрезмерное его увеличение приводит к эффекту гиперразмерности — потере обобщающих свойств сети, выражающейся в возрастании ошибки обобщения (тестирования).

Этап 4. Обучение сети

Обучение сети — очень важный, но не окончательный этап создания нейросетевой математической модели. Цель обучения — подобрать синаптические веса w_{ij} так, чтобы на каждый входной вектор $\boldsymbol{X}_q$ множества обучающих примеров сеть выдавала вектор $\boldsymbol{Y}_q$, минимально отличающийся от заданного выходного вектора $\boldsymbol{D}_q$. Эта цель достигается путем использования ***алгоритмов обучения нейронной сети***.



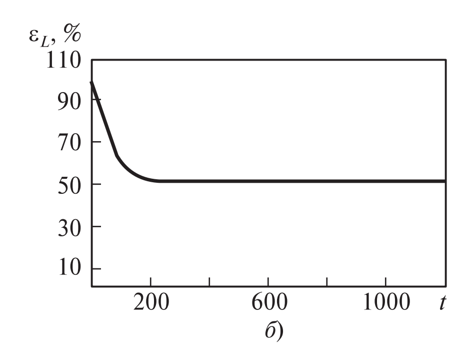


Рис. 6.18. Характерные кривые зависимости ошибки обучения ε_L от числа эпох t в случае, когда сеть обучается успешно (*а*) и когда процесс обучения не дает желаемого результата (*б*)

Характерная *кривая обучения* — зависимость ошибок обучения от количества эпох обучения — приведена на рис. 6.18, *а*. Однако может случиться, что сеть «не захочет» обучаться: ошибка обучения с увеличением количества эпох не будет стремиться к нулю (см. рис. 6.18, *б*). Причинами этого нежелательного явления могут быть следующие.

1-я причина. Недостаточное количество скрытых слоев и скрытых нейронов — тогда рекомендуется увеличить их количество.

2-я причина. Наличие в обучающем множестве *противоречащих* друг другу (*конфликтных*) примеров, когда одним и тем же наборам входных параметров соответствуют разные наборы выходных параметров. Например, одним и тем же симптомам соответствуют разные диагнозы заболеваний. Обнаружить такие примеры в обучающем множестве можно путем его визуального анализа или путем применения специальных программ. Затем следует разобраться в причинах возникновения конфликтных примеров: некоторые из них могут просто оказаться ошибочными, и их нужно удалить. Другая причина может быть связана с тем, что в самой структуре входного вектора отсутствуют какие-то параметры (например, возраст больного, рост, вес, цвет его глаз и др.), также оказывающие влияние на диагноз. В этом случае рекомендуется вернуться на этап 1 алгоритма (см. рис. 6.17) и пересмотреть постановку задачи, увеличить размерность входного вектора $\boldsymbol{X}$, добавив дополнительные параметры, которые своими значениями обеспечат непротиворечивость примеров обучающего множества.

3-я причина. Попадание *в локальный минимум*. Эта проблема связана с тем, что поверхность функции-ошибки персептрона, схематично изображенная на рис. 6.6, имеет достаточно сложный характер со множеством мелких «ямочек», называемых *локальными минимумами*. Процесс обучения персептрона состоит в движении по этой

поверхности небольшими шагами в сторону антиградиента, т. е. в сторону наибольшего наклона поверхности функции-ошибки. Естественно, что, опускаясь таким образом по поверхности функции-ошибки, можно «застрять» в каком-либо мелком локальном минимуме, не достигнув самого глубокого минимума, называемого *глобальным*. В этом случае рекомендуется попробовать заново начать процесс обучения из другой начальной точки или сменить алгоритм обучения нейросети.

4-я причина. Наличие в множестве примеров поведения предметной области *посторонних выбросов* — примеров, которые не подчиняются закономерностям предметной области и значительно отличаются по своим параметрам от другой статистической информации, например, вследствие закравшихся ошибок. Рекомендуется выявить такие примеры, например, с помощью методики [57] (см. § 6.5), проанализировать их на предмет наличия ошибок и, если таковые обнаружатся — исправить ошибки или удалить выбросы.

5-я причина. *Паралич* сети. В этом случае процесс обучения «замирает» в результате неуправляемого возрастания синаптических весов и соответствующего возрастания аргументов активационных функций. Следствием является попадание в область насыщения, где производные от сигмоидных функций близки к нулю. Рекомендуется использовать активационные функции, которые не имеют горизонтальных асимптот, например логарифмические (см. § 6.5). Другой вариант — попробовать провести нормализацию чисел, формирующих примеры предметной области.

6-я причина. При слишком большой скорости обучения может потеряться устойчивость итерационного процесса, тогда рекомендуется уменьшить ее.

Заметим, что в практике нейросетевого моделирования обычно случается так, что не удается установить точную причину отсутствия сходимости процесса обучения нейронных сетей. Остается только предполагать эти причины и поочередно принимать меры по устранению возникших нежелательных явлений, т. е. выполнять рекомендации, указанные в приведенных выше шести пунктах.

Этап 5. Проверка и оптимизация сети

Проверка обобщающих свойств сети (иногда данный этап называют *тестированием* сети) производится на тестирующем множестве примеров, т. е. на тех примерах, которые не были использованы при обучении сети. Результаты тестирования полезно представить графически в виде гистограммы, на которой значения желаемых выходов персептрона $\boldsymbol{D}_q$ сопоставлены с действительными (прогнозными) $\boldsymbol{Y}_q$ — теми, которые вычислил персептрон. Пример такой гистограммы приведен на рис. 6.19.

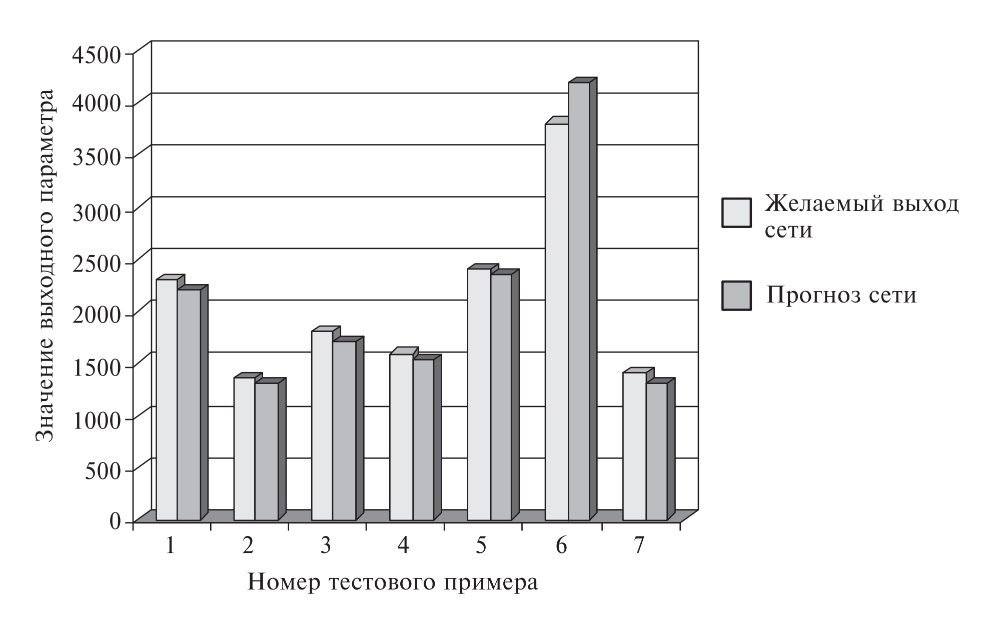


Рис. 6.19. Пример гистограммы, показывающей соотношение желаемых выходов сети с действительными (прогнозными)

Если разница между компонентами желаемого выходного вектора $\boldsymbol{D}_q$ и действительного выходного вектора $\boldsymbol{Y}_q$ окажется незначительной, то можно переходить к следующему, этапу **6**, не выполняя *оптимизацию* сети. Однако, чтобы лишний раз убедиться в адекватности разрабатываемой нейросетевой математической модели, полезно вернуться на этап **2** и те примеры, которые были тестирующими (либо часть тестирующих примеров), включить в обучающее множество, а часть примеров, бывших обучающими, сделать тестирующими. После этого снова повторить этапы **3**, **4** и **5**.

Если же погрешность обобщения сети окажется неприемлемо большой, то надо попытаться *оптимизировать* сеть. Эта операция состоит в подборе наиболее подходящей для данной задачи структуры сети — количества скрытых слоев, количества скрытых нейронов, количества синаптических связей, а также вида и параметров активационных функций нейронов. В некоторых нейропакетах предусмотрена автоматическая оптимизация сети, но бывает полезно выполнить такую оптимизацию вручную, построив график зависимости погрешности обобщения ε_T от числа скрытых нейронов (см. рис. 6.2) и от других параметров персептрона, а затем с помощью этих графиков выбрать структуру сети, обеспечивающую минимальную погрешность обобщения.

При выборе оптимальной структуры сети следует помнить, что цель оптимизации сети состоит в минимизации погрешности обоб-

щения ε_T, но не погрешности обучения ε_L. Именно по величине погрешности обобщения судят о качестве сети, о ее обобщающих и, следовательно, прогностических свойствах. Погрешность же обучения — это всего лишь промежуточный результат. Желательно, чтобы она была небольшой, но добиваться самого минимального ее значения вовсе не обязательно и даже вредно, поскольку, как видно из графиков рис. 6.2, это приводит к эффекту переобучения, т. е. росту погрешности обобщения.

Понятно, что оптимизация нейронной сети подразумевает многократные возвраты назад на этапы **4**, **3**, **2** или даже на этап **1**. На этапе **1** заново выполняется постановка задачи, включающая переоценку значимости входных параметров с последующим их сокращением, или, наоборот, добавлением. Напомним, что малозначимые входные параметры могут быть выявлены с помощью разрабатываемой нейросети путем поочередного исключения входных параметров и наблюдением за погрешностью обобщения. Если при исключении какого-либо входного параметра погрешность обобщения нейронной сети возрастет, то этот параметр является значимым для данной математической модели. В противном случае параметр не является значимым, и его не следует учитывать при постановке задачи.

После оптимизации сети ее обобщающие свойства рекомендуется проверить на примерах подтверждающего множества $\boldsymbol{P}$. Дело в том, что в процессе оптимизации сеть может «приспособиться» к примерам тестирующего множества. Если эти примеры по каким-либо причинам не характерны для всей предметной области, то на других примерах, которых не было ни в тестирующем, ни в обучающем множествах, она может дать неожиданно большую ошибку. Для исключения такого явления, а также чтобы окончательно убедиться в адекватности разрабатываемой математической модели, вычисляют ошибку *прогнозирования* сети ε_P на подтверждающем множестве, т. е. на тех примерах, которые не участвовали ни в обучении, ни в тестировании.

Результатом оптимизации и проверки сети является готовая к использованию *нейросетевая математическая модель* предметной области — *интеллектуальная информационная система* (ИИС).

Этап 6. Исследование предметной области — интеллектуальный анализ данных

Путем проведения вычислительных экспериментов над математической нейросетевой моделью достигаются цели моделирования и находятся ответы на поставленные вопросы. Например, могут быть решены такие задачи, как оптимизация моделируемого объекта, прогнозирование его будущих свойств, выявление закономерностей предметной области и др.

Нейросетевая математическая модель, если она правильно спроектирована и обучена, «впитала в себя» закономерности моделируемой предметной области. Она реагирует на изменение входных параметров и ведет себя точно так же, как вела бы себя сама предметная область. Поэтому над такой моделью надо поставить как можно больше экспериментов. Можно попробовать поменять некоторые из входных параметров и посмотреть, как при этом меняются значения выходного вектора $\boldsymbol{Y}$. Например, меняя возраст кандидата в президенты, можно пронаблюдать за изменением его рейтинга, как это было сделано в § 5.6. Надо постараться извлечь из этих виртуальных экспериментов как можно больше полезной информации.

Часто бывает полезно графически изображать зависимости выходных параметров модели от входных. Например на рис. 5.6 были построены зависимости рейтингов Д. А. Медведева и В. В. Жириновского от их возраста, а на рис. 5.7 — гистограмма, позволяющая давать практические рекомендации по повышению политического рейтинга И. М. Хакамады. Эти интересные результаты и выводы были получены путем исследования предметной области (политического процесса — выборов президента страны) с помощью нейросетевой математической модели. Результаты моделирования полезно представлять в виде объемных фигур, как например на рис. 5.8.

Выполняя исследования предметной области с помощью нейросетевой математической модели, следует понимать, что во многих задачах входные параметры обычно коррелированны между собой. Например, при прогнозировании здоровья пациента на будущие периоды времени (см. § 5.2) недостаточно изменять только один входной параметр, отвечающий за его возраст, поскольку с возрастом обязательно изменяются другие его параметры — содержание холестерина и сахара в крови, артериальное давление и др.

Кроме того, следует помнить, что нейронные сети часто выявляют так называемые «ложные корреляционные зависимости». По этому поводу уместно привести классический пример. Однажды методом интеллектуального анализа статистической информации было установлено, что люди, злоупотребляющие кофе, чаще других болеют онкологическими заболеваниями. В то же время, известно, что кофе не содержит канцерогенных веществ. Объяснение этого парадокса нашлось, когда обнаружили, что за чашкой кофе часто следовала выкуренная сигарета. Она-то и явилась причиной онкозаболеваний.

В заключение укажем, что алгоритм создания интеллектуальной системы и ее применения для интеллектуального анализа данных, приведенный на рис. 6.17, апробирован при решении многих практических задач и представляется вполне эффективным. Однако в каждом конкретном случае, а также в зависимости от опыта, навыков и предпочтений разработчика интеллектуальной системы возможны

отклонения от этого алгоритма. Например, если у разработчика хорошо развита интуиция или наработаны собственные алгоритмы для определения количества нейронов в скрытых слоях, то совсем не обязательным является использование формулы Арнольда — Колмогорова — Хехт-Нильсена.

Контрольные вопросы и задания к § 6.6

1. Перечислите этапы создания нейросетевой математической модели предметной области.
2. Как формируется структура входного вектора $\boldsymbol{X}$ и выходного вектора $\boldsymbol{D}$?
3. Перечислите способы формирования содержимого пар векторов $\boldsymbol{X}_q$–$\boldsymbol{D}_q$.
4. Как выполняется проектирование сети?
5. В чем состоит цель обучения сети и как она достигается?
6. Назовите шесть причин, по которым сеть может не поддаваться обучению.
7. В чем состоит цель тестирования сети?
8. В чем состоит цель оптимизации сети?
9. В каких случаях используется подтверждающее множество примеров, и можно ли без него обойтись?
10. Приведите примеры ложных корреляционных зависимостей.
11. Зайдите на сайт **www.LbAi.ru**, скачайте и выполните все лабораторные работы. При выполнении лабораторных работ полезно использовать книги [73, 88].
12. Придумайте тему своего собственного проекта, лежащую в области ваших личных интересов. Выполните проект, следуя пунктам алгоритма, приведенного на рис. 6.17 (см. § 6.6), и используя программный инструментарий для работы с нейронными сетями, скачанный с сайта **www.LbAi.ru** (Лабораторная работа № 8). При выполнении задании полезно использовать книги [73, 88].
13. Перечислите свойства, которые нейросети и нейрокомпьютеры унаследовали от своего прототипа — человеческого мозга. Проявились ли некоторые из этих свойств при выполнении вашего проекта?

Глава 7
НЕКЛАССИЧЕСКИЕ НЕЙРОННЫЕ СЕТИ

Нейронные сети персептронного типа с сигмоидными активационными функциями, предложенные, исследованные и развитые в классических работах У. Мак-Каллока и В. Питтса [100], Ф. Розенблатта [101, 102], Б. Уидроу и М. Е. Хоффа [107], А. И. Галушкина с соавторами [6–11] и др., являются в настоящее время наиболее изученными и наиболее применяемыми из всего множества разновидностей известных нейронных сетей. Однако развитие нейроинформатики на этой нейросетевой парадигме не остановилось. Новейшие исследования мозга фиксируют все новые и новые данные, которые выходят за рамки классических гипотез нейроинформатики. Соответственно, появляются новые (неклассические) нейросетевые парадигмы, некоторые из них рассматриваются в настоящей главе.

7.1. СЕТЬ КАСКАДНОЙ КОРРЕЛЯЦИИ ФАЛЬМАНА–ЛИБЬЕРА

В работе *С. Е. Фальмана* и *К. Либьера* [91] предложен оригинальный способ постепенного наращивания структуры нейронной сети в процессе ее обучения. Идея авторов состоит в следующем. Сначала создается простейший однослойный персептрон, затем, после попытки его обучения, следует серия этапов, состоящих в том, что в сеть добавляется по одному скрытому нейрону, каждый из которых своими синапсами подключается ко всем входным нейронам сети и ко всем ранее добавленным скрытым нейронам. Синаптические веса каждого скрытого нейрона подбираются так, чтобы обеспечивался максимальный коэффициент корреляции между его активностью (т. е. силой выходного сигнала) и ошибкой обучения сети, вычисляемой с помощью формулы:

$$S = \sum_{i=1}^{I} \left| \sum_{q=1}^{Q} \left(y^{(q)} - \overline{y}\right) \left(\varepsilon_i^{(q)} - \overline{\varepsilon_i}\right) \right|, \tag{7.1}$$

где I — количество выходных нейронов; Q — количество обучающих примеров; $y^{(q)}$ — выходной сигнал нейрона-кандидата на q-м обучающем примере; $\varepsilon_i^{(q)} = d_i^{(q)} - y_i^{(q)}$ — погрешность i-го выходного нейрона

на q-м обучающем примере; $\overline{y}$ и $\overline{\varepsilon_i}$ — средние значения соответственно y и ε_i, рассчитанные по всему множеству обучающих примеров.

По мнению авторов нейронной сети, таким способом обеспечивается максимальная востребованность вновь добавляемого скрытого нейрона.

Затем аксон скрытого нейрона подключается к выходным нейронам сети и продолжается ее обучение, в процессе которого корректируются только синапсы выходных нейронов сети. Поскольку в обучении сети синаптические веса скрытых нейронов не корректируются, то в использовании алгоритма обратного распространения ошибки необходимости нет, и сеть обучают, например, с помощью обобщенного дельта-правила.

В результате выполнения нескольких таких этапов, сопровождаемых каждый раз добавлением нового скрытого нейрона, формируется нейронная сеть оптимальной структуры.

Таким образом, алгоритм создания сети каскадной корреляции (так ее назвали авторы) выглядит следующим образом.

Шаг 1. Формируется нейросеть (рис. 7.1), состоящая из одних только входных и выходных нейронов — однослойный персептрон. Число входных и выходных нейронов персептрона определено условиями задачи. На рисунке 7.1 сеть имеет три входных и два выходных нейрона, хотя их количество может быть произвольным.

Шаг 2. Однослойный персептрон обучается обычным способом, например, с помощью обобщенного дельта-правила.

Шаг 3. Добавляется скрытый нейрон-кандидат (рис. 7.2). Его си-

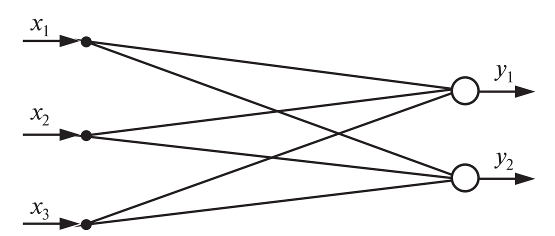


Рис. 7.1. Формируется однослойный персептрон

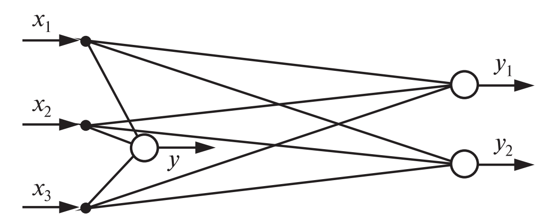


Рис. 7.2. Добавляется скрытый нейрон-кандидат, который своими синапсами подключается ко всем входным нейронам сети, а его аксон остается свободным

напсы подключаются ко всем входным нейронам сети, а аксон пока остается свободным. Далее следует процесс обучения скрытого нейрона. Оно состоит в подборе его синаптических весов из условия обеспечения максимального коэффициента корреляции между активностью скрытого нейрона-кандидата, определяемой его выходным сигналом, и значением погрешности на выходе сети. Скрытый нейрон-кандидат обучается с использованием того же множества обучающих примеров, которое использовалось при обучении исходного однослойного персептрона.

Шаг 4. Нейрон-кандидат включается в существующую структуру сети — его аксон подключается к выходным нейронам (рис. 7.3) и возобновляется процесс обучения сети, причем синаптические веса скрытого нейрона больше не модифицируются. Если при обучении целевая функция (квадратичная погрешность сети) снижается до заданного предела, то процесс обучения завершается. В противном случае выполняется следующий шаг.

Шаг 5. Добавляется новый скрытый нейрон-кандидат, который своими синапсами подключается ко всем входным нейронам и к уже существующим скрытым нейронам (рис. 7.4), а его аксон остается свободным. Скрытый нейрон-кандидат обучается с использованием обучающего множества примеров. Цель обучения, как и ранее, состоит в подборе синаптических весов скрытого нейрона-кандидата, при

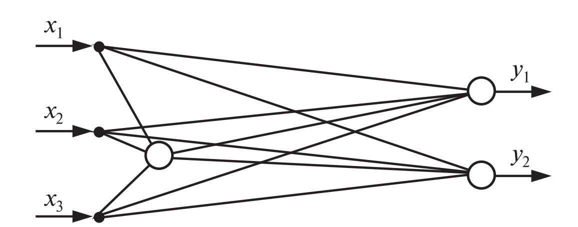


Рис. 7.3. Скрытый нейрон подключается ко всем выходным нейронам сети

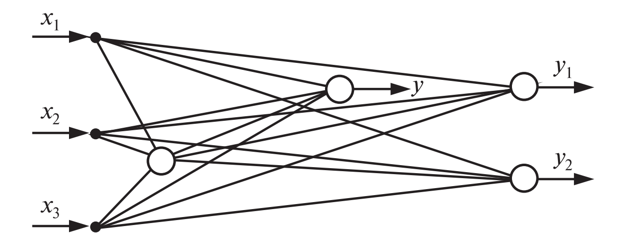


Рис. 7.4. Добавляется новый скрытый нейрон-кандидат, который своими синапсами подключается ко всем входным нейронам сети и к скрытому нейрону (скрытым нейронам), а его аксон остается свободным

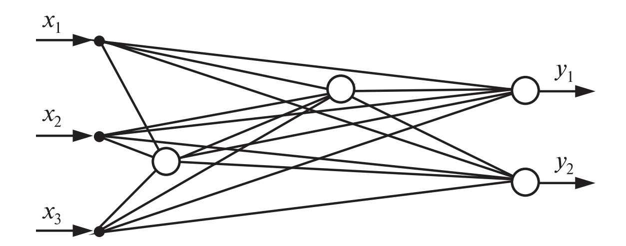


Рис. 7.5. Скрытый нейрон включается в существующую структуру сети

котором корреляция между его активностью и значением погрешности на выходе сети становится максимальной.

Шаг 6. Скрытый нейрон-кандидат включается в существующую структуру сети (рис. 7.5), возобновляется процесс обучения сети, причем синаптические веса скрытых нейронов больше не модифицируются. Если при обучении целевая функция (квадратичная погрешность сети) снижается до заданного предела, то процесс обучения завершается. В противном случае добавляется новый скрытый нейрон и т. д.

Как видим, в результате выполнения алгоритма Фальмана—Либьера формируется нейронная сеть, структура которой отличается от слоистой, и поэтому сеть не является персептроном.

Существуют модификации алгоритма, отличающиеся тем, что на каждом шаге добавляется и обучается не один скрытый нейрон-кандидат, а сразу несколько, из которых потом выбирается лучший, обладающий наибольшим коэффициентом корреляции. Как отмечают авторы алгоритма [91], параллельное корреляционное обучение нескольких нейронов-кандидатов и выбор наилучшего из них уменьшают вероятность попадания в точку локального минимума и ввода в сеть скрытого нейрона с плохо подобранными весами.

К достоинствам сети каскадной корреляции можно отнести следующие:

1) поскольку в процессе минимизации целевой функции (квадратичной погрешности сети) задействованы только синаптические веса выходных нейронов, то не требуется использование алгоритма обратного распространения ошибок;

2) оптимизация структуры сети выполняется автоматически в ходе выполнения пунктов алгоритма создания сети;

3) как отмечается многими специалистами, алгоритм каскадной корреляции демонстрирует прекрасные качества как средство обучения и построения нейронных сетей оптимальной структуры.

7.2. РАДИАЛЬНО-БАЗИСНЫЕ СЕТИ

Радиально-базисные сети (RBF-сети) были предложены в 1985 г. британским математиком *М. Пауэллом* (M. J. D. Powel). В их основе лежит понятие радиально базисной-функции (RBF-функции). ***RBF-функцией*** называется функция, радиально изменяющаяся вокруг некоторого центра, заданного вектором $\boldsymbol{C}$, и принимающая ненулевые значения только в окрестности этого центра. За пределами этой окрестности значения RBF-функции равны нулю либо стремятся к нулю. Аргументом RBF-функции является расстояние между текущим вектором $\boldsymbol{X}$ и вектором $\boldsymbol{C}$, т. е. $\phi = \phi(\|X - C\|)$.

Заметим, что $\boldsymbol{X}$ и $\boldsymbol{C}$ — векторы, поэтому радиально-базисная функция является функцией многих аргументов. При $\boldsymbol{C} = 0$ $\phi = \phi(\|X\|) = \phi(\sqrt{x_1^2 + x_2^2 + \ldots})$. А если входной вектор $\boldsymbol{X}$ вырожден в скаляр x, то $\phi = \phi(\sqrt{x^2}) = \phi(|x|)$. Это значит, что функция ϕ является четной, т. е. симметричной относительно вертикальной оси, как, например, показано на рис. 7.6.

Если вектор $\boldsymbol{X}$ имеет две компоненты x_1 и x_2, то RBF-функция может иметь вид, как на рис. 7.7.

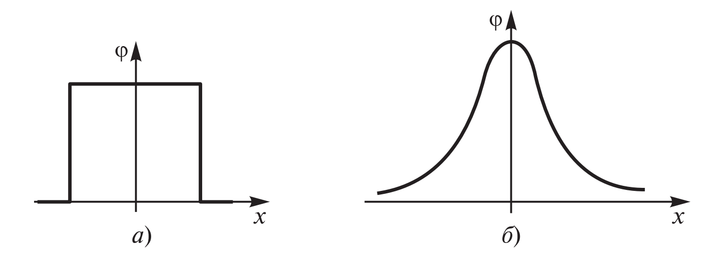


Рис. 7.6. Примеры радиально-базисных функций одного аргумента

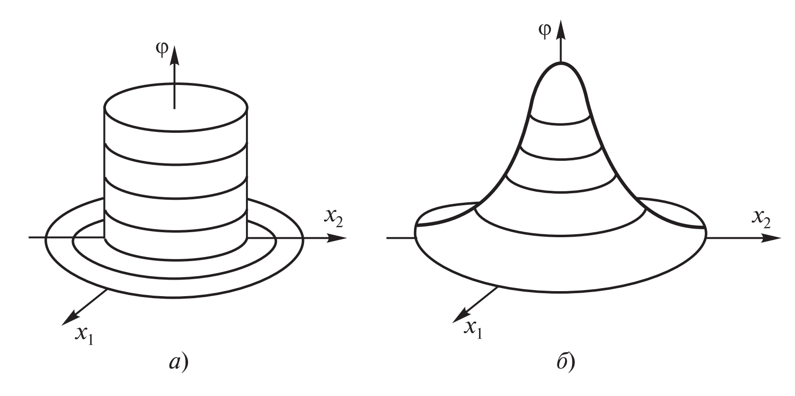


Рис. 7.7. Примеры радиально-базисных функций двух аргументов

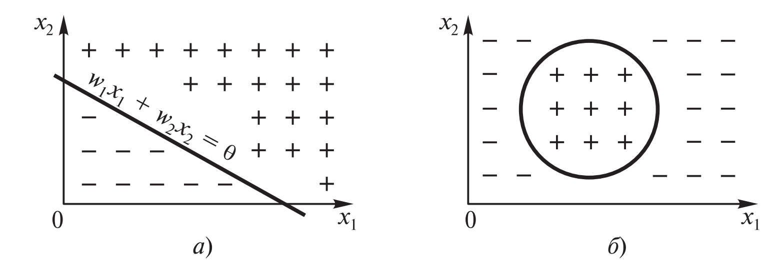


Рис. 7.8. Разделение пространства на две части нейроном Мак-Каллока—Питтса (*а*) и радиально-базисным нейроном (*б*)

В некотором смысле нейроны, имеющие радиально-базисные активационные функции, являются логическим дополнением нейронов со ступенчатыми и сигмоидными активационными функциями. Действительно, нейрон Мак-Каллока—Питтса активизируется, когда $\sum_i x_i w_i - \theta \geqslant 0$, т. е. он имеет единичный (либо положительный) выход для точек пространства, лежащих по одну сторону гиперплоскости $\sum_i x_i w_i - \theta = 0$, и нулевой (либо отрицательный) выход для точек, лежащих по другую сторону (рис. 7.8, *а*). Нейрон с радиально-базисной функцией также делит пространство входных параметров на две части, однако разделяющей поверхностью здесь является гиперсфера (рис. 7.8, *б*). Для точек пространства, лежащих внутри гиперсферы, выход нейрона положителен, а для точек, лежащих снаружи гиперсферы, он равен нулю (либо приближается к нулю).

Интуитивно ясно, что радиально-базисные нейроны обладают преимуществом. Оно заключается в том, что с их помощью легче построить поверхность, обеспечивающую разделение входных параметров на классы. В связи с этим в радиально-базисных сетях нет необходимости использовать большое количество скрытых слоев. Так, типичная радиально-базисная сеть имеет только один скрытый слой, причем синаптические веса нейронов скрытого слоя равны единице, а нейроны входного и выходного слоев имеют линейные активационные функции. Как доказано в [41], такая сеть при достаточном количестве нейронов скрытого слоя гарантирует решение любой задачи классификации образов.

Рассмотрим RBF-сеть, изображенную на рис. 7.9, которая имеет N входов, один выход и J радиально-базисных нейронов скрытого слоя. Множество обучающих примеров для такой сети состоит из Q входных N-мерных векторов $\boldsymbol{X}_q$, $q = 1, 2, \ldots, Q$ и соответствующих им выходных скаляров d_q. Заметим, что в общем случае нейронов выходного слоя может быть много, так что на выходе сети может быть не скаляр (d_q), а вектор ($\boldsymbol{D}_q$). Однако пока мы ограничимся рассмотрением RBF-сетей только с одним выходным нейроном.

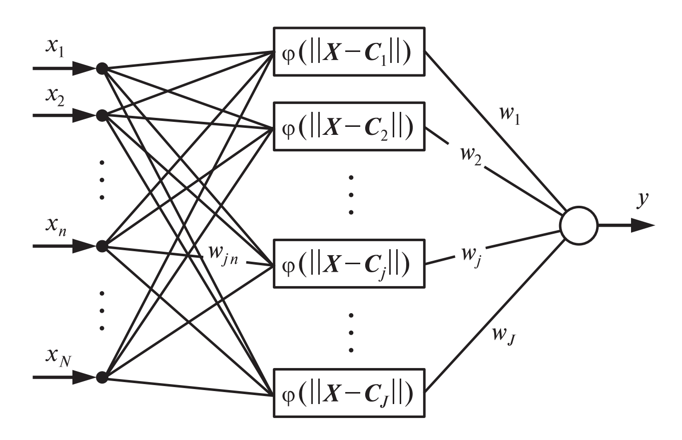


Рис. 7.9. Радиально-базисная нейронная сеть

Если число нейронов скрытого слоя J задать равным числу обучающих примеров Q, то работу RBF-сети, преобразующей входные векторы $\boldsymbol{X}_q$ в выходные скаляры d_q, можно представить с помощью матричной операции:

$$\begin{bmatrix} \phi(\|\boldsymbol{X}_1-\boldsymbol{C}_1\|) & \phi(\|\boldsymbol{X}_1-\boldsymbol{C}_2\|) & \ldots & \phi(\|\boldsymbol{X}_1-\boldsymbol{C}_j\|) & \ldots & \phi(\|\boldsymbol{X}_1-\boldsymbol{C}_J\|) \\ \phi(\|\boldsymbol{X}_2-\boldsymbol{C}_1\|) & \phi(\|\boldsymbol{X}_2-\boldsymbol{C}_2\|) & \ldots & \phi(\|\boldsymbol{X}_2-\boldsymbol{C}_j\|) & \ldots & \phi(\|\boldsymbol{X}_2-\boldsymbol{C}_J\|) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi(\|\boldsymbol{X}_q-\boldsymbol{C}_1\|) & \phi(\|\boldsymbol{X}_q-\boldsymbol{C}_2\|) & \ldots & \phi(\|\boldsymbol{X}_q-\boldsymbol{C}_j\|) & \ldots & \phi(\|\boldsymbol{X}_q-\boldsymbol{C}_J\|) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi(\|\boldsymbol{X}_Q-\boldsymbol{C}_1\|) & \phi(\|\boldsymbol{X}_Q-\boldsymbol{C}_2\|) & \ldots & \phi(\|\boldsymbol{X}_Q-\boldsymbol{C}_j\|) & \ldots & \phi(\|\boldsymbol{X}_Q-\boldsymbol{C}_J\|) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_j \\ \cdots \\ w_J \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdots \\ d_q \\ \cdots \\ d_Q \end{bmatrix},$$

которую перепишем в сокращенной матричной форме:

$$\boldsymbol{\Phi w} = \boldsymbol{d}. \tag{7.2}$$

Искомый вектор весовых коэффициентов получается отсюда простым обращением матрицы радиально-базисных функций:

$$\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\boldsymbol{d}. \tag{7.3}$$

Таким образом, для обучения RBF-сети не требуется итерационного процесса!

К сожалению, этот результат представляет только теоретический интерес. Практической же ценности он не имеет. Дело в том, что при большом числе обучающих примеров требование $J = Q$ приводит к необходимости проведения сложных вычислений из-за чрезмерного увеличения числа нейронов внутреннего слоя. Кроме того, создаваемая нейросетью гиперповерхность, аппроксимирующая точки предметной области, в точности проходит через точки, изображающие обучающие примеры. Как отмечалось ранее (см. § 6.1), такая ситуация, называемая переобучением, или гиперразмерностью, ослабляет обобщающие

свойства сети. Поэтому в практически используемых RBF-сетях число нейронов скрытого слоя выбирают значительно меньше числа обучающих примеров, т. е. $J << Q$. В этом случае матрица $\mathbf{\Phi}$ не является квадратной, так как количество строк Q в ней значительно больше количества столбцов J, т. е. в системе (7.2) количество уравнений больше количества неизвестных. Поэтому весовые коэффициенты w_j приходится определять не из системы уравнений (7.2), а другим способом, например из условия минимума квадратичной ошибки сети

$$\boldsymbol{\varepsilon} = \sum_{q=1}^{Q}\left[\sum_{j=1}^{J} w_j \boldsymbol{\phi}\left(\|\boldsymbol{X}_q - \boldsymbol{C}_j\|\right) - d_q\right]^2. \qquad (7.4)$$

В отличие от многослойного персептрона функция ошибки RBF-сетей (7.4) не имеет локальных минимумов. Это значит, что при обучении RBF-сетей проблемы обхода локальных минимумов не существует. Задача минимизации функции ошибки (7.4) является линейной, и поэтому здесь применимы хорошо известные методы линейной оптимизации, которые сходятся на порядок быстрее, чем в случае обучения многослойного персептрона. Более того, задача определения весов w может быть решена путем псевдоинверсии прямоугольной матрицы $\mathbf{\Phi}$:

$$w = \mathbf{\Phi}^{+}\boldsymbol{d}, \qquad (7.5)$$

где $\mathbf{\Phi}^{+} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{T}$.

В качестве активационных функций в RBF-сетях чаще всего применяется функция Гаусса.

$$\phi(\|\boldsymbol{X} - \boldsymbol{C}_j\|) = \exp\left(-\frac{\|\boldsymbol{X} - \boldsymbol{C}_j\|^2}{2\sigma_j^2}\right). \qquad (7.6)$$

Вид этой функции определяется двумя параметрами: вектором $\boldsymbol{C}_j$, задающим ее центр, и скаляром σ_j^2, задающим скорость ее убывания с ростом евклидова расстояния между центром $\boldsymbol{C}_j$ и текущей координатой $\boldsymbol{X}$. Как видно на рис. 7.10, параметром σ_j^2 задается расстояние от оси ординат до точки перегиба кривой Гаусса. Иногда этот параметр называют *шириной окна* функции Гаусса.

В случае, когда число нейронов скрытого слоя равно числу обучающих примеров ($J = Q$), центры активационных функций логично задать координатами векторов обучающей выборки, т. е. $\boldsymbol{C}_j = \boldsymbol{X}_q$

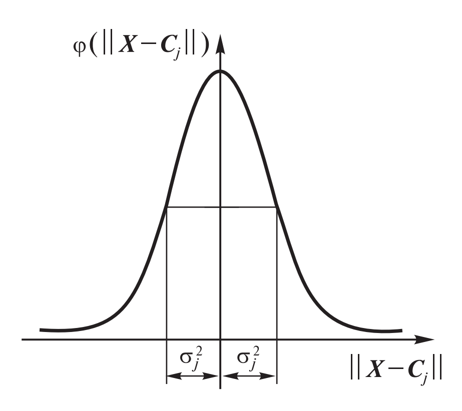


Рис. 7.10. Функция Гаусса: σ_j^2 — расстояние от оси ординат до точки перегиба функции

($j = q = 1, \ldots, Q$), а ширины окон σ_j^2 подобрать из тех соображений, чтобы часть пространства, в которой располагаются векторы $\boldsymbol{X}_q$, была охвачена влиянием активационных функций. Например, σ_j^2 можно задать как евклидово расстояние от центра $\boldsymbol{C}_j$ до его ближайшего соседа

$$\sigma_j^2 = \|\boldsymbol{C}_j - \boldsymbol{C}_k\|, \tag{7.7}$$

где k — номер ближайшего соседа.

Иногда σ_j^2 задают как среднее расстояние до P ближайших соседей:

$$\sigma_j^2 = \frac{1}{P}\sum_{p=1}^{P}\|\boldsymbol{C}_j - \boldsymbol{C}_p\|, \tag{7.8}$$

где P рекомендуется задавать в интервале $[3, 5]$.

После этого определяют весовые коэффициенты $\boldsymbol{w}$.

В реальной же ситуации $J \ll Q$, поэтому предварительно выполняют *кластеризацию* — объединяют близкие между собой векторы $\boldsymbol{X}_q$ в *кластеры*. Затем определяют центры кластеров, в которые и помещают центры активационных функций. Один из способов кластеризации векторов на заданное количество кластеров рассмотрен в § 7.5.

После фиксации центров активационных функций находят ширины их окон σ_j^2, например, по формуле (7.7). Последний этап состоит в определении синаптических весов w путем минимизации функционала (7.4) или с помощью вычисления по формуле (7.5).

В заключение отметим, что в последнее время RBF-сети начинают успешно применять для решения задач, которые традиционно решались нейронными сетями с сигмоидными функциями. В основном это задачи распознавания и классификации, задачи аппроксимации функций и задачи прогнозирования. Интерес к RBF-сетям объясняется их следующими достоинствами:

1) RBF-сети имеют всего один скрытый слой, что избавляет конструктора сети от решения вопроса о числе слоев;

2) обучение RBF-сети сводится к решению линейной оптимизационной задачи, поэтому отсутствует опасность попадания в локальный минимум, а сам процесс обучения занимает на порядок меньше времени, чем процесс обучения многослойного персептрона.

Однако при проектировании RBF-сетей приходится решать вопрос об оптимальном количестве нейронов скрытого слоя, выполнять кластеризацию входных обучающих векторов и определять ширины окон активационных функций.

Многие исследователи склонны полагать, что изобретение RBF-сетей является одним из наиболее крупных достижений в области нейроинформатики, и ставят это изобретение в один ряд с открытием алгоритма обратного распространения ошибки или изобретением генетических алгоритмов.

7.3. РЕКУРРЕНТНЫЕ СЕТИ НА БАЗЕ ПЕРСЕПТРОНА

Как показали нейрофизиологические исследования, мозг человека имеет гораздо более сложную структуру и механизмы взаимодействия нейронов, чем те, которые реализованы в рассмотренных выше искусственных нейронных сетях. В частности, между биологическими нейронами выявлено большое количество не только прямых, но и *обратных* связей. Поэтому были предприняты попытки дополнить искусственные нейронные сети обратными связями, что привело к неожиданным новым результатам. Рассмотрим некоторые из них.

На рисунке 7.11, *а* дан пример персептрона, у которого выходные сигналы y_1 и y_2 через элементы единичных задержек z^{-1} подаются обратно на входы персептрона. Таким образом, под воздействием входных сигналов x_1 и x_2 на выходе сети в момент времени t вырабатываются сигналы $y_1(t)$ и $y_2(t)$, а затем, в следующий момент времени $t+1$, под воздействием этих сигналов, подаваемых на вход через линии задержки z^{-1}, вырабатываются новые выходные сигналы $y_1(t+1)$ и $y_2(t+1)$.

Нетрудно показать, что для всякой рекуррентной сети может быть построена идентичная сеть без обратных связей с прямым распространением сигнала (рис. 7.11, *б*), поэтому для обучения рекуррентных сетей может быть применен метод обратного распространения ошибки.

В настоящее время нашли применение рекуррентные нейросети, в которых элементы единичных задержек включены как в обрат-

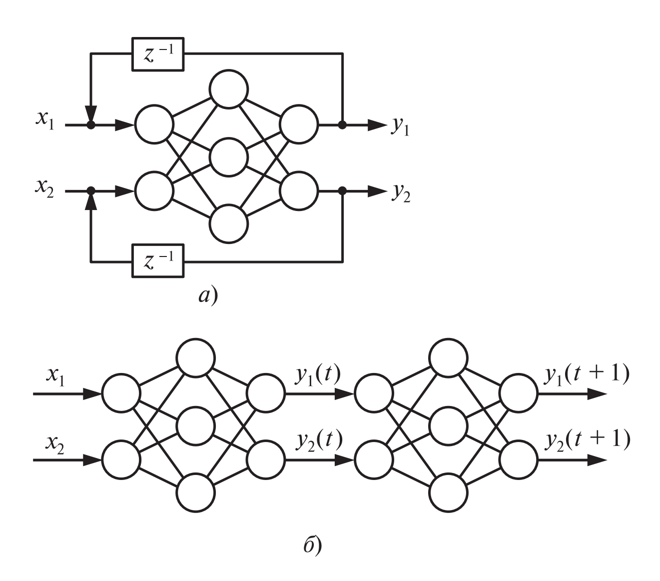


Рис. 7.11. Рекуррентная сеть на базе персептрона (*а*) и идентичный ей развернутый персептрон (*б*)

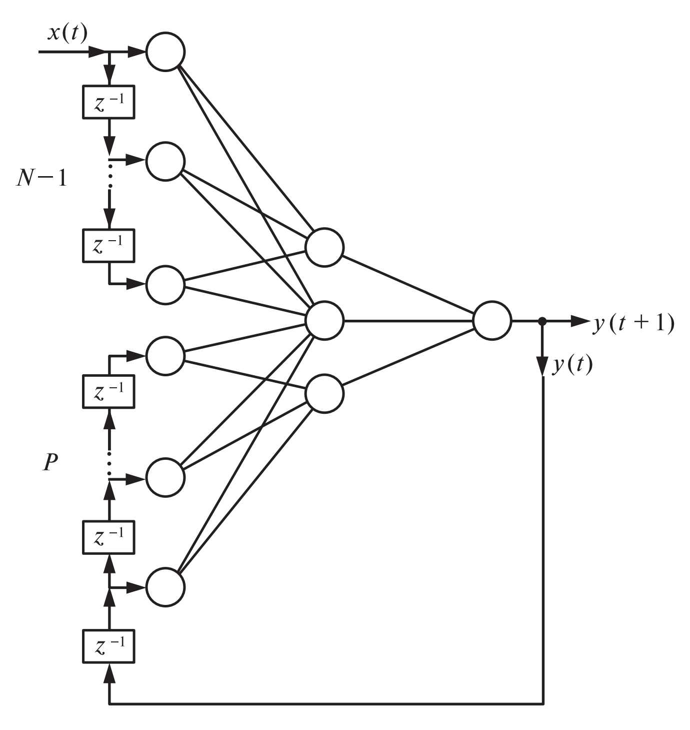


Рис. 7.12. Рекуррентная сеть с $N-1$ элементами задержки входного сигнала и P элементами задержки в обратной связи

ные, так и во входные связи, а сами обратные связи исходят как от выходных нейронов, так и от нейронов скрытых слоев. На рисунке 7.12 приведена сеть, имеющая один вход и один выход, причем как входной, так и выходной сигналы подаются на нейроны скрытого слоя через элементы задержек. Таким образом, выходной сигнал, образующийся в момент времени $t+1$, является функцией $N+P$ переменных:

$$y(t{+}1)=f\,(x(t), x(t-1), \ldots, x(t-(N-1)), y(t-1), y(t-2), \ldots, y(t-P)), \quad (7.9)$$

из которых N переменных представляют собой последовательность входных сигналов, а P переменных являются ответами персептрона в разные моменты времени. Их называют *контекстными* аргументами.

Такие нейронные сети удобно использовать для прогнозирования временных рядов. Например, если речь идет о задаче прогнозирования курса американского доллара, рассмотренной в [63], то в скользящем окне в качестве дат можно рассматривать N входных аргументов, а в качестве соответствующих значений курса доллара — P контекстных аргументов формулы (7.9). Шириной скользящего окна будет

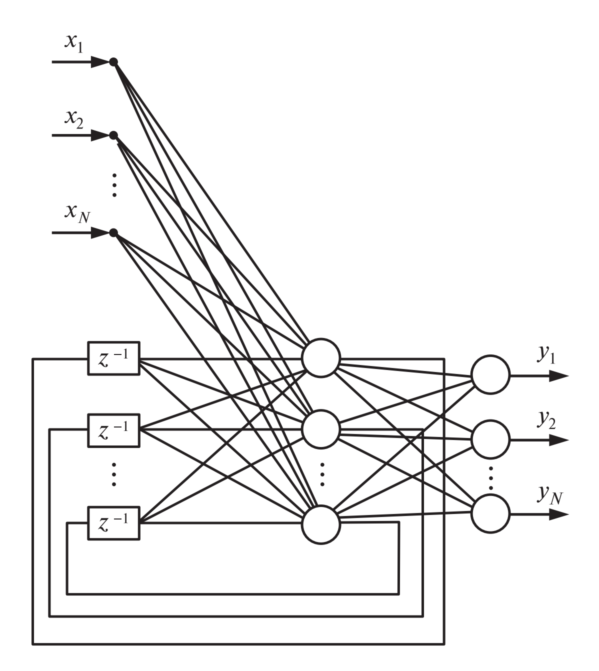


Рис. 7.13. Структура нейронной сети Элмана

величина N, которую следует принять равной P. Значение функции $y(t+1)$ тогда будет означать прогноз курса доллара на день вперед.

Если курсы валют требуется прогнозировать с учетом влияния большего количества различных факторов, то следует воспользоваться рекуррентной сетью с несколькими выходами. Таким свойством обладает, например, сеть Элмана [41], структура которой приведена на рис. 7.13.

Рекуррентные сети рассмотренного типа применяются также для математического моделирования динамических объектов. В этом случае уточнение весов выступает в роли идентификации параметров динамической математической модели. Созданная таким образом математическая модель динамического объекта может применяться для управления данным объектом: машиной, устройством, развивающимся во времени процессом.

7.4. РЕКУРРЕНТНАЯ СЕТЬ ХОПФИЛДА

Американский физик *Джон Хопфилд* [98] обратил внимание на то, что динамический процесс, возникающий в замкнутой на себя рекуррентной сети, может привести к некоторому устойчивому состоянию, отличающемуся от исходного. Другими словами, итерационный процесс рекуррентной сети может вывести ее на стационарный режим,

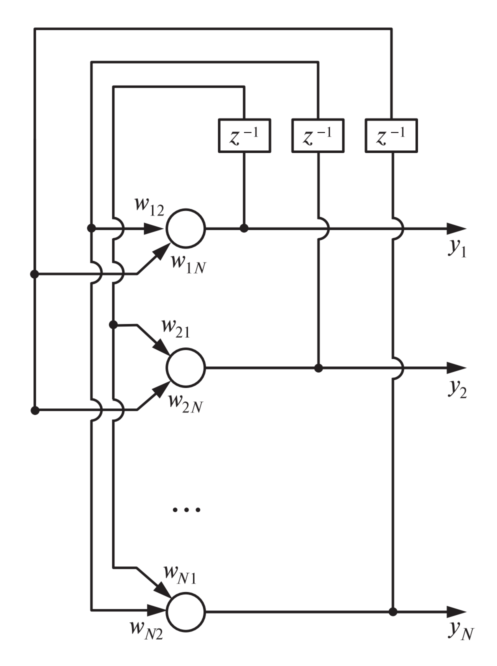


Рис. 7.14. Структура нейронной сети Хопфилда

при котором состояние сети перестанет меняться. Причем это конечное стационарное состояние сети зависит как от ее первоначального состояния, так и от значений элементов матрицы синаптических весов.

Сеть Хопфилда в классическом варианте исполнения приведена на рис. 7.14. Считается, что она не имеет входных элементов, а входной вектор задает первоначальную активность нейронов, которая затем изменяется в ходе итерационного процесса, обусловленного наличием обратных связей. В процессе итераций активность нейронов корректируется с помощью формул:

$$S_i = \sum_{j=1,\ i \neq j}^{N} w_{ij} y_j(t), \tag{7.10}$$

$$y(t+1) = \begin{cases} 1, & \text{если } S_i \geqslant 0; \\ -1, & \text{если } S_i < 0, \end{cases} \tag{7.11}$$

т. е. принимает значения либо +1, либо −1. Согласно схеме рис. 7.14, в ней отсутствуют связи нейронов с их собственными выходами.

Сеть работает следующим образом. Сначала входной вектор задает начальную активность $y_i(t)$ каждого нейрона. Затем выбранный случайным образом нейрон получает взвешенные сигналы от всех остальных нейронов и обновляет свое состояние согласно формулам (4.79)–(4.80). Выбирается следующий нейрон, и процесс повторяется

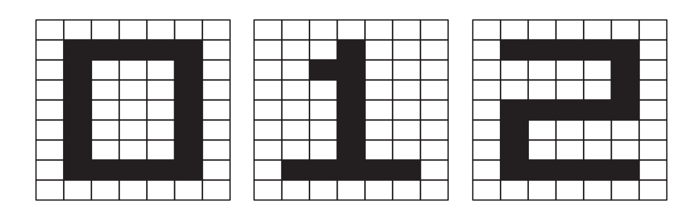


Рис. 7.15. Три образа, запомненные сетью Хопфилда

до тех пор, пока нейроны, выбранные для обновления, не перестанут изменять свое состояние. Наступает стационарный режим.

Сеть Хопфилда ведет себя подобно памяти, хранящей заданный заранее набор образов и пытающейся вспомнить один из них, если ей предъявляется какой-либо из этих образов, искаженный помехами. Образы предварительно кодируются в виде векторов с бинарными компонентами. Каждый из векторов умножается сам на себя, образуя квадратную матрицу. Затем матрицы складываются, образуя новую квадратную матрицу, главная диагональ которой обнуляется. Это и есть матрица синаптических весов w_{ij}, хранящая информацию о всех заданных образах.

Для пояснения алгоритма формирования матрицы синаптических весов приведем пример. Допустим, что предметная область содержит два образа, закодированных с помощью двух векторов: $[-1, 1, -1]$ и $[1, -1, 1]$. Умножая их самих на себя и складывая, получим квадратную матрицу

$$\begin{bmatrix}-1\\1\\-1\end{bmatrix}\begin{bmatrix}-1 & 1 & -1\end{bmatrix}+\begin{bmatrix}1\\-1\\1\end{bmatrix}\begin{bmatrix}1 & -1 & 1\end{bmatrix}=\begin{bmatrix}1 & -1 & 1\\-1 & 1 & -1\\1 & -1 & 1\end{bmatrix}+\begin{bmatrix}1 & -1 & 1\\-1 & 1 & -1\\1 & -1 & 1\end{bmatrix}=\begin{bmatrix}2 & -2 & 2\\-2 & 2 & -2\\2 & -2 & 2\end{bmatrix}.$$

Выполнив обнуление главной диагонали, окончательно получим:

$$w_{ij}=\begin{bmatrix}0 & -2 & 2\\-2 & 0 & -2\\2 & -2 & 0\end{bmatrix}.$$

Теперь предположим, что мы закодировали и ввели в сеть Хопфилда матрицу синаптических весов, соответствующую трем образам, изображенным на рис. 7.15. После этого мы предъявляем сети входной вектор (т. е. задаем первоначальную активность нейронов), соответствующий некоторому искаженному образу, изображенному на рис. 7.16 в левом верхнем углу. Как видно из последующих кадров рис. 7.16, итерационный процесс привел к тому, что на выходе сети Хопфилда сформировался вектор, в точности соответствующий одному из ранее введенных образов. В этом случае говорят, что входной образ *ассоциировался* с одним из введенных ранее образов и что рекуррентные сети рассмотренного типа выступают в роли *ассоциативных* запоминающих устройств.

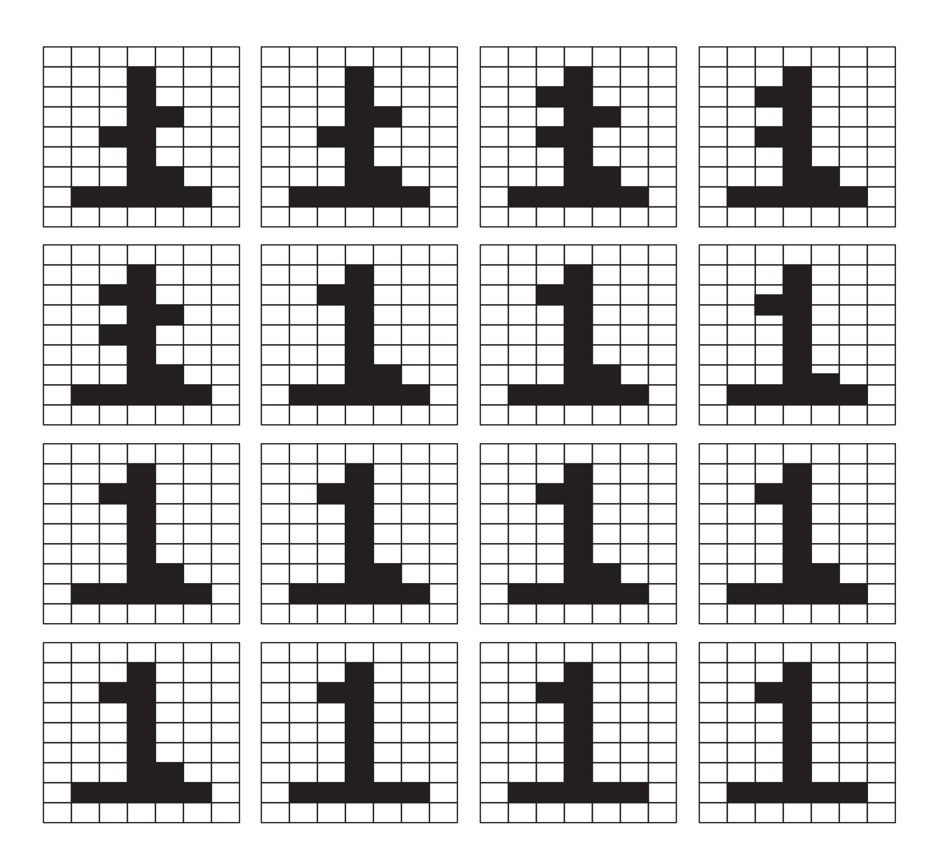


Рис. 7.16. Предъявленный нейронной сети Хопфилда искаженный образ и его деформация за последующие 15 итераций (материал заимствован из [26])

Хопфилду удалось математически строго показать, что при любом входном векторе итерационный процесс всегда приведет к распознаванию одного из введенных ранее образов, наиболее близкому к предъявленному. Однако максимальное количество запоминаемых сетью образов $p_{\max}$ ограничено формулой [98]:

$$p_{\max} = \frac{N}{2 \ln N},$$

где N — число нейронов сети Хопфилда.

7.5. САМООБУЧАЮЩИЕСЯ И ГИБРИДНЫЕ СЕТИ

Рассмотренные выше нейронные сети персептронного типа обучались путем тренировки на примерах. Для обучения предоставлялась первоначальная информация о предметной области в виде набора входных векторов $\boldsymbol{X}_q$ и им соответствующих выходных векторов $\boldsymbol{D}_q$ — свое-

го рода подсказок, с использованием которых сеть обучалась давать правильные ответы на задаваемые вопросы. Поэтому такой способ называют *обучением с учителем*.

В реальных условиях любой живой организм, взаимодействуя с окружающей средой, постоянно ощущает ее воздействие, получает своего рода подсказки, согласно которым корректирует свое поведение. Так, кошка, один раз прыгнувшая на раскаленную печь, никогда больше не повторит своей ошибки. Подобных примеров можно привести тысячи, поэтому механизм обучения с учителем, несомненно, свойственен мозгу любого живого существа.

Однако мозг человека обладает и другими механизмами обучения. Мы в состоянии решать многие интеллектуальные задачи и без предварительного обучения. Например, мы не задумываясь можем выполнить кластеризацию объектов — объединить похожие между собой объекты в отдельные классы, называемые кластерами.

Рассмотрим нейронную сеть, которая без помощи учителя автоматически настраивает свои синаптические веса, решая задачу кластеризации входных векторов. Сеть, изображенная на рис. 7.17, имеет один слой I нейронов, каждый из которых соединен с N входами. Этот слой нейронов называют *слоем Кохонена* в честь ученого, предложившего алгоритм самообучения нейросети. Веса синаптических связей каждого i-го нейрона слоя Кохонена образуют вектор синаптических связей $\boldsymbol{W}_i = [w_{i1}, w_{i2}, \ldots, w_{iN}]^T$, размерность которого совпадает с размерностью входных векторов $\boldsymbol{X} = [x_1, x_2, \ldots, x_N]^T$. Первоначально значения компонент векторов $\boldsymbol{W}_i$ задаются датчиком случайных чисел, а компоненты входного вектора $\boldsymbol{X}$ подвергаются нормализации путем деления каждой из них на длину самого вектора $\boldsymbol{X}$.

Между вектором $\boldsymbol{X}$ и каждым из векторов $\boldsymbol{W}_i$ вычисляются евклидовы расстояния:

$$\|X - W_i\| = \sqrt{\sum_{n=1}^{N} (x_n - w_{in})^2}, \tag{7.12}$$

среди которых выбирается наименьшее. Нейрон, у которого вектор синаптических весов $\boldsymbol{W}_i$ оказался ближе всего к входному вектору $\boldsymbol{X}$, будем называть *нейроном-победителем*, а его номер обозначим через w, т. е. нейрон-победитель имеет порядковый номер $i = w$. Синаптические веса нейрона-победителя, а также всех близлежащих к нему нейронов подвергаются корректировке по формуле Кохонена

$$\boldsymbol{W}_i\,(t+1) = \boldsymbol{W}_i\,(t) + \eta_i(t)\,G(i,w)\,(\boldsymbol{X} - \boldsymbol{W}_i\,(t))\,. \tag{7.13}$$

В этой формуле коэффициент скорости обучения $\eta_i(t)$ уменьшается с увеличением времени t, а функция $G(i, w)$ зависит от расстояния

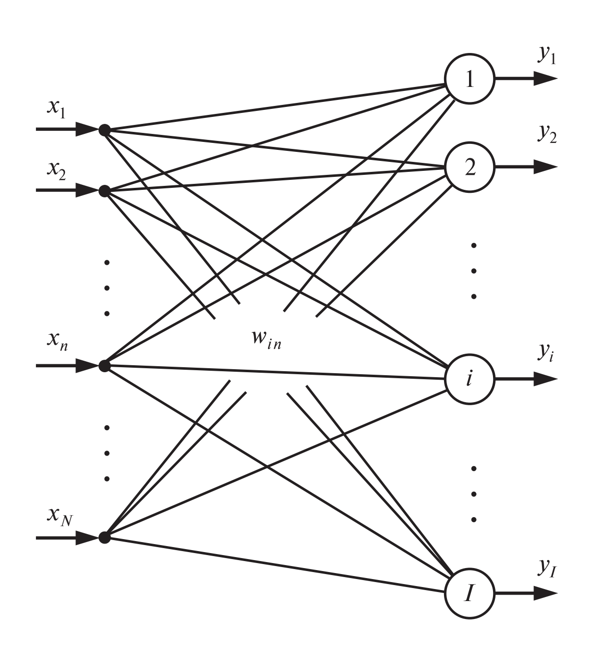


Рис. 7.17. Самообучающаяся нейронная сеть

между нейроном-победителем w и i-м нейроном сети. Обычно ее задают максимальной для $i = w$ и убывающей по мере увеличения расстояния от i-го нейрона до нейрона-победителя. Таким свойством обладает, например, функция Гаусса

$$G(i, w) = \exp\left(-\frac{\|\boldsymbol{W}_w - \boldsymbol{W}_i\|^2}{2\lambda^2}\right), \tag{7.14}$$

в которой коэффициент λ называется *уровнем соседства*.

В другом варианте обучению по формуле Кохонена (7.13) подвергается только нейрон-победитель. В этом случае функция $G(i, w)$ имеет вид

$$G(i, w) = \begin{cases} 1, & \text{для } i = w; \\ 0, & \text{для } i \neq w. \end{cases} \tag{7.15}$$

Этот алгоритм называется алгоритмом WTA. Название образовано начальными буквами английских слов *Winner Takes All*, что в переводе означает «Победитель забирает все». В отличие от этого алгоритма формулу (7.14) относят к алгоритмам типа WTM — *Winner Takes Most,* что переводится как «Победитель забирает больше».

Согласно итерационной формуле Кохонена (7.13), нейрон-победитель на каждой эпохе приближает свой синаптический вектор $\boldsymbol{W}_w$ к входному вектору $\boldsymbol{X}$, как проиллюстрировано на рис. 7.18. В результате такого обучения каждой отдельной группе близких между собой входных векторов $\boldsymbol{X}_q$, называемой *кластером*, будет соответствовать

один-единственный нейрон. Этот нейрон в ходе обучения для этих векторов был победителем, причем его синаптический вектор в результате итерационного процесса (7.13) окажется в центре этого кластера.

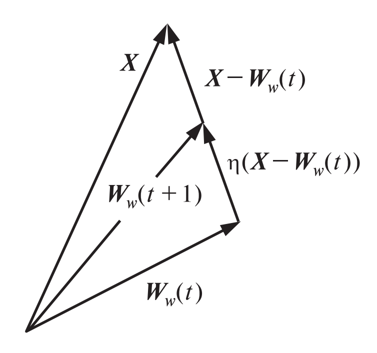


Рис. 7.18. Нейрон-победитель приближает свой синаптический вектор $\boldsymbol{W}_w$ к входному вектору

Однако могут найтись и такие нейроны, которые ввиду их первоначальной удаленности от входных векторов так и не были ни разу победителями. Этим нейронам не будет соответствовать ни один кластер, поэтому их называют *мертвыми* нейронами. Наличие мертвых нейронов нежелательно, так как они снижают эффективность вычислительного алгоритма.

Проблема мертвых нейронов решается путем введения механизма, моделирующего *эффект утомления*, известный из нейрофизиологических наблюдений. Этот эффект заключается в том, что биологические нейроны сразу после победы на некоторое время теряют свою активность и не участвуют в конкурентной борьбе.

Существует несколько способов моделирования эффекта утомления нейронов, например путем введения *потенциала активности* каждого нейрона. Потенциалы активности модифицируются всякий раз после представления очередного входного вектора:

$$p_i(t+1) = \begin{cases} p_i(t) - p_{\min}, & \text{для } i = w; \\ p_i(t) + \frac{1}{N}, & \text{для } i \neq w. \end{cases} \tag{7.16}$$

В (7.16) N — количество нейронов нейросети; $p_{\min}$ — минимальное значение потенциала, разрешающее нейрону участие в конкурентной борьбе, обычно принимаемое $p_{\min} \approx 0,75$. Если текущее значение потенциала какого-либо нейрона падает ниже $p_{\min}$, то этот нейрон временно отдыхает, уступая место для конкурентной борьбы другим нейронам.

В результате такого обучения весовые векторы всех нейронов распределятся так, что они будут центрами различных кластеров, образованных из входных векторов, причем число кластеров, на которые разобьется входное множество векторов, будет равно числу нейронов сети.

Рассмотренная нейронная сеть представляет практический интерес, поскольку с помощью ее можно решать задачи кластеризации объектов, которые возникают, например, при необходимости сжатия информации с сохранением глобальных свойств сжимаемого множества.

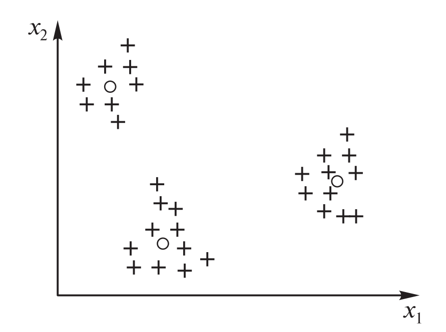


Рис. 7.19. Пример кластеризации множества: крестиками обозначены входные параметры, кружочками — центры кластеров

На рисунке 7.19 приведен результат работы нейронной сети, которая разбила исходное множество на три кластера.

Задачи кластеризации возникают в самых разнообразных областях человеческой деятельности. Так, в педагогике часто возникает задача разделения всех учащихся на несколько классов, например вундеркиндов, отличников, успевающих и неуспевающих. Исходной информацией для такого деления является множество различных показателей успеваемости учащихся за длительный период времени. Для решения этой задачи можно использовать слой Кохонена из четырех нейронов.

Слой Кохонена эффективно использовать в сочетании со слоями нейронов, реализующими другие нейросетевые парадигмы. Так, применение RBF-сетей, рассмотренных в § 7.2, подразумевает обязательную предварительную кластеризацию входных векторов, которую удобно выполнять с помощью нейронного слоя Кохонена. На рисунке 7.20 изображена гибридная сеть, содержащая слой Кохонена, выходные сигналы с которого передаются на вход обычного персептрона. Обучение гибридной нейросети осуществляется в две стадии.

На первой стадии происходит самообучение слоя Кохонена, в результате которого множество входных векторов разбивается на кластеры. Число кластеров равно числу нейронов I слоя Кохонена, а векторы синаптических весов каждого нейрона этого слоя принимают значения, изображающиеся центрами образовавшихся кластеров. Теперь при подаче на вход сети какого-либо входного вектора каждый нейрон слоя Кохонена будет вырабатывать сумму $S_i = \sum\limits_{n=1}^{N} w_{in}x_n$. Эти суммы нормализуются так, чтобы выходной сигнал нейрона-победителя был равен единице, а выходные сигналы остальных нейронов принимали значения в интервале (0;1). Указанная операция нормализации может осуществляться, например, с помощью формулы:

$$y_i = \exp\left(-\frac{(S_{\max} - S_i)^2}{\sigma^2}\right),$$

в которой значение параметра σ подбирается индивидуально для решаемой задачи.

Персептронная часть сети обучается обычным способом, например методом обратного распространения ошибки. Обучающая выборка в этом случае состоит из выходных векторов слоя Кохонена $\boldsymbol{Y}_q$ и со-

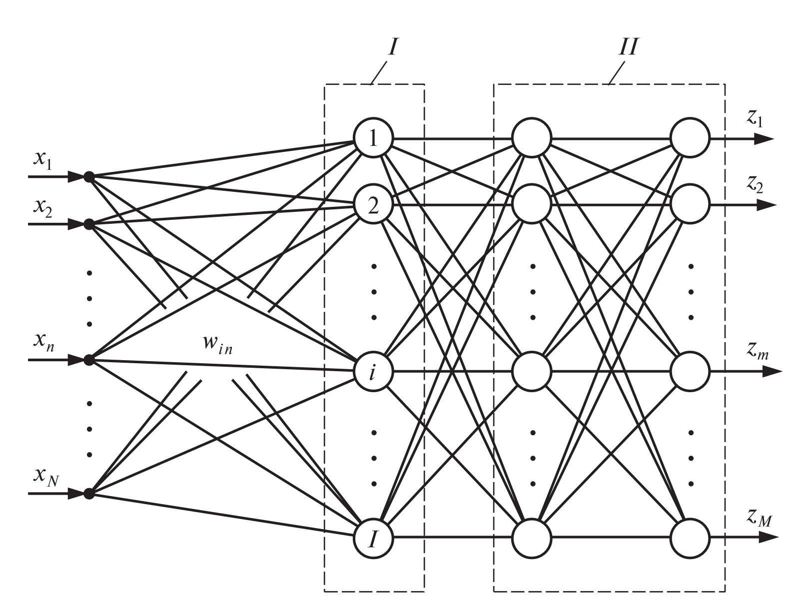


Рис. 7.20. Гибридная нейронная сеть: *I* — слой нейронов Кохонена; *II* — персептронные слои

ответствующих им желаемых выходов сети $\boldsymbol{D}_q$. Благодаря хорошему структурированию исходных данных, выполненному слоем Кохонена, персептронная часть гибридной сети обучается во много раз быстрее, чем обычный персептрон.

В заключение отметим, что, согласно современным данным, мозг человека представляет собой каскадное объединение биологических нейронных сетей различных функциональных назначений. Поэтому гибридная сеть, объединяющая слои нейронов различных нейросетевых парадигм, в большей мере соответствует современным представлениям о структуре и способе функционирования мозга. Следствием такого объединения является ее более высокая эффективность.

7.6. ИНТЕЛЛЕКТУАЛЬНЫЙ НЕЙРОН ВАЛЬЦЕВА

Согласно сложившейся классификации [33], все рассмотренные выше парадигмы нейронных сетей относятся к так называемому *коннекционистскому* (от англ. *connect* — соединяться) типу. Согласно этому подходу, нейронные сети строятся на основе относительно простых математических моделей нейронов, например математических нейронов Мак-Каллока — Питтса. Все же многообразие необходимых свойств и качеств нейронных сетей достигается не путем усложнения матема-

тических нейронов, а путем их *соединения* в определенным образом спроектированные структуры.

Здесь мы рассмотрим другой альтернативный тип нейронных сетей, называемый *нейробиологическим*. Этот тип отличается использованием более сложных математических моделей нейронов, которые, по мнению их авторов, более полно отражают свойства биологических нейронов мозга.

В настоящее время доминирующим в искусственном интеллекте является коннекционистский подход. В первую очередь это обусловлено достаточно хорошей формализацией моделей. Однако, как только мы переходим от задач аппроксимации к решению сложных многоуровневых задач, требующих привлечения семантики, оказывается, что необходимо вводить функциональные элементы более сложные, нежели формальные нейроны. Попытки «выращивания» подобного рода сложных нейронов из сети простых приводят к тому, что количество формальных нейронов катастрофически растет и появляются неустранимые сложности обучения таких сетей.

В 1996 г. российским ученым-нейробиологом В. Б. Вальцевым с соавторами [105] была предложена математическая модель *большого пирамидального нейрона мозга*. Как известно, пирамидальные нейроны составляют до 80% коры головного мозга. Они отличаются тем, что их тело похоже на пирамиду, из которой выходит один большой дендрит, который ветвится на множество мелких шипов-дендритов. Через дендриты пирамидальный нейрон получает сигналы от тысяч других клеток.

Схема математического нейрона Вальцева изображена на рис. 7.21. Основными его функциональными элементами являются тело клетки, в котором происходит накопление потенциала от входов, и аксонный холмик, который генерирует на выходе нейрона нервные импульсы, когда потенциал клетки превысит некоторое пороговое значение. Та-

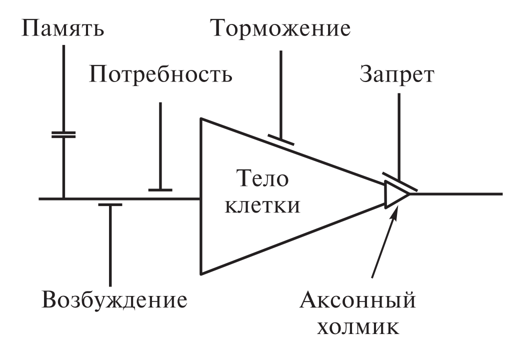


Рис. 7.21. Структура нейрона Вальцева

ким образом, нейрон Вальцева, как и математический нейрон Мак-Каллока — Питтса, может находиться в двух состояниях: возбужденном и невозбужденном.

Математический нейрон Вальцева имеет один выход и несколько входов.

Вход запрета выполняет роль тормоза аксонного холмика. Под действием сигналов входа запрета передача импульсов с нейрона может быть прекращена, при этом потенциал клетки не изменится.

При подаче импульсов на *вход торможения* уменьшается потенциал клетки, т. е. снижается активность нейрона.

При подаче импульсов на *вход потребности* (*регуляции*) потенциал клетки нарастает, но, при наличии импульсов только на этом входе, не может превысить его порогового значения, необходимого для генерации выходных импульсов.

При наличии импульсов только на входе *возбуждения* потенциал клетки не может превысить значение, равное 30% от порогового.

Вход памяти является обучаемым. При подаче импульсов на максимально обученный вход памяти, потенциал клетки не сможет подняться выше значения в 70% от порогового. В состоянии нулевой обученности импульсы, подаваемые на вход памяти не увеличивают потенциала клетки.

Нейрон Вальцева выполняет суммирование потенциалов входов:

$$\phi = \phi_0 + \sum_{n=1}^{4} \phi_n,$$

где ϕ_0 — начальный потенциал, ϕ_n — потенциалы входов памяти, регуляции, возбуждения и торможения соответственно при $n = 1, \ldots, 4$.

Когда потенциал ϕ превысит пороговое значение ψ и при этом потенциал на входе запрета не превысит некое заданное значение, нейрон переходит в возбужденное состояние и генерирует электрические импульсы. Частота импульсов прямо пропорциональна превышению потенциала ϕ над порогом ψ:

$$\omega = \Omega(\phi - \psi) + \omega_0,$$

где Ω — коэффициент зависимости частоты генерации от надпорогового потенциала.

Как можно заметить, в отличие от классических математических нейронов, в которых все входы являются равноценными, в нейроне Вальцева различные входы по-разному влияют на поведение нейрона. Так, при появлении импульсов на входе торможения, нейрон Вальцева перестает генерировать выходные сигналы независимо от того, какие сигналы он получает с других входов, и независимо от потенциала клетки.

Возникает естественный вопрос: когда это может быть полезно?

Это может быть полезно, например, в задаче управления поведением кибернетического объекта, когда окружающая среда, в которой действует объект, характеризуется множеством сигналов, поступающих с различных датчиков, выдающих сигналы разной степени важности. Например, если речь идет об управлении движением самолета, то сигнал, поступающий с датчика высоты, при ее уменьшении ниже критической имеет первостепенное значение, требующее принятия экстренных мер вне зависимости от сигналов, поступающих с других датчиков.

Таким образом можно констатировать, что нейронная сеть нейробиологического типа, в отличие от классической коннекционистской нейросети, обладает дополнительными свойствами, характерными для экспертной системы, а именно, свойством принимать решения, руководствуясь еще и наперед заданными правилами, например: «Экстренно запустить программу набора высоты при снижении высоты полета самолета ниже критического значения».

Другой, более сложный пример применения нейронов Вальцева приведен в работе [105]. В этой статье рассматривается задача создания системы управления для робота-эксурсовода. Робот-экскурсовод должен двигаться по определенному маршруту — линии с нанесенными на нее метками. Достигнув очередной метки, робот должен останавливаться и читать небольшую лекцию.

Основным требованием к роботу-экскурсоводу является безопасность. Робот ни в коем случае не должен причинять вред людям и экспонатам и должен немедленно останавливаться в случае возникновения какого-либо препятствия или при потере маршрутной линии. Для этого он оснащен датчиками препятствий двух видов: первого уровня, определяющими препятствие непосредственно в зоне безопасности робота, и второго уровня, реагирующими на удаленные препятствия. Реакцией на сигнал датчика первого уровня должна быть немедленная и безусловная остановка. Реакция на сигнал датчика второго уровня — менее критична. При поступлении этого сигнала робот может оценивать расстояние до препятствия и планировать возможность его объезда.

Помимо всего прочего, робот должен следить за уровнем заряда своих бортовых аккумуляторов. При уменьшении заряда ниже критического робот должен начать процедуру поиска источника питания.

Понятно, что сигналы, которые должен обрабатывать робот, имеют разные приоритеты, поэтому при разработке системы управления роботом-экскурсоводом лучше использовать нейробиологические нейроны, которые как раз и приспособлены для обработки сигналов, имеющих разные приоритеты.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Можно ли назвать нейронную сеть Фальмана—Либьера персептроном? Почему?
2. Назовите преимущества и недостатки нейронной сети Фальмана—Либьера по сравнению с персептроном.
3. Назовите преимущества и недостатки радиально-базисной нейронной сети по сравнению с персептроном.
4. Назовите преимущества и недостатки нейронных сетей Эльмана, Хопфилда и Кохонена по сравнению с персептроном.
5. В чем состоит разница между нейроном Мак-Каллока—Питтса и нейроном Вальцева?

Глава 8

КОМПЬЮТЕРНОЕ ТВОРЧЕСТВО

8.1. ФИЛОСОФСКИЕ АСПЕКТЫ ТВОРЧЕСТВА

Любое произведение искусства может быть закодировано в виде конечного числа цифр. Например, каждое слово поэмы состоит из букв, которые могут быть закодированы 33 цифрами. Ясно, что при таком соответствии одна длинная строка цифр может рассматриваться как кодированная запись поэмы.

Аналогично обстоит дело в живописи. Полотно картины можно расчертить на мельчайшие клетки и цвет каждой клетки закодировать цифрами. То же самое в музыке. Из анализа Фурье известно, что все звучание музыкального произведения, от первой ноты до последней, может быть представлено одной-единственно кривой на экране осциллографа. Кривую можно с любой степенью точности кодировать цифрами.

Таким образом, любое произведение искусства в любой области можно представить в виде набора конечного числа цифр. Число возможных комбинаций этих цифр огромно, но не бесконечно. Поэтому можно представить себе библиотеку, содержащую все возможные комбинации цифр. Подавляющее большинство комбинаций цифр в переводе на буквы, цвета и звуки не имеют никакого смысла. Но среди этих комбинаций есть и такие, которые имеют смысл и которые являются тем, что мы называем произведениями искусства. Существуют ли алгоритмы, которые позволят компьютеру выбрать из множества бессмысленных вариаций те, которые являются гениальными поэмами, картинами, симфониями?

Первые попытки создания таких алгоритмов относятся к XVII в. Известен, например, пятисотстраничный трактат немецкого иезуита *Афанасиуса Кирхера* «Универсальная музургия, или великое искусство созвучий и диссонансов». А. Кирхер был учеником Р. Луллия и рассматривал музыкальную композицию как комбинаторную задачу. Его идеи были реализованы в виде устройства, напоминающего механическую экспертную систему Луллия (см. § 1.1), которое ныне хранится в музее Кембриджского университета.

В начале XVIII в. вопросами механического сочинения музыкальных произведений с помощью таблиц и игральных костей занимались многие известные композиторы, такие как Бах, Гайдн, Моцарт.

Рассмотрим вопрос создания произведений искусства с применением известных нам современных методов искусственного интеллекта. Мы

уже сталкивались с методами математического моделирования в различных естественных науках, таких как физика, метеорология, экономика, механика сплошных сред, электроника и пр. Роль математического моделирования в жизни современной цивилизации переоценить трудно, причем круг проблем, решаемых этим методом, постоянно растет.

Напомним, что модель — это «черный ящик», в который вводятся входные и выводятся выходные параметры. Модель является намеренно упрощенной схемой некоторого реального объекта, системы, процесса. Но путем исследования модели получают рекомендации для решения реальных проблем.

Математическая модель может существовать в виде логических программ, переводимых на язык ЭВМ. Математическую модель, введенную в компьютер, называют компьютерной моделью.

Существуют общие принципы построения моделей. Приведем некоторые из них. Для построения модели необходимо:

1) выявить релевантные (существенные) факторы, т. е. факторы, которые могут сказываться на результатах решения данной проблемы или на исходе рассматриваемого процесса;

2) выбрать факторы, которые могут быть описаны количественно;

3) объединить факторы по общим признакам и сократить их перечень, выявить инварианты (о них речь пойдет дальше);

4) установить количественные соотношения между выбранными факторами и инвариантами.

Факторы, которые по своей природе не могут быть выражены количественно, так же как и несущественные факторы, исключаются из рассмотрения.

При математическом моделировании очень важным этапом является установление инвариантов системы. Идея инвариантности состоит в том, что, хотя система в целом претерпевает последовательные изменения, некоторые ее свойства сохраняются неизменными. Существование инварианта в любом множестве неизбежно влечет за собой ограничение разнообразия. Существование инварианта во множестве явлений говорит об ограничении разнообразия. Поэтому теория инвариантов — это теория ограничения разнообразия.

Поскольку любой закон природы подразумевает существование некоторого инварианта, то, следовательно, всякий закон природы есть ограничение разнообразия, а так как цель науки — поиск законов, то наука занимается поиском ограничений разнообразия.

В математике *инвариантом* называют функцию от преобразуемой величины, не изменяющую своего значения при преобразовании этой величины. Так, расстояние между двумя точками является инвариантным относительно переноса начала координат и поворота осей на любой угол. В гидродинамике инвариантами являются критерии подобия — числа Грасгофа, Прандтля, Релея и т. д.

В лингвистике выделяют инварианты стихотворных форм. Например, старинная итальянская форма стиха — сонет — всегда имеет 14 строк. Первая часть его состоит из двух катренов (четверостиший), вторая — из двух терцетов (трехстиший). Стихотворный размер сонета — пятистопный (реже шестистопный) ямб. Форма рифмовки для катренов — две разнозвучные рифмы, а для терцетов — две рифмы, отличающиеся от рифм катренов. Третья строка первого терцета рифмуется со второй строкой второго терцета и т. д. В любом сонете легко обнаружить сохранение указанных инвариантов.

Аналогичным образом инварианты могут быть обнаружены и в других произведениях искусства. Возможность создания тех или иных произведений искусства может быть определена в первом приближении как сознательная или бессознательная способность находить нужные инварианты и комбинировать их для получения желаемого эффекта. Эта способность проявляется художником-творцом в результате обобщения закономерностей всего предшествующего художественного наследия.

Таким образом, мы видим, что искусство в конечном итоге преследует ту же цель, что и наука, — ***выявление инвариантов, установление связи между ними, ограничение разнообразия***.

О возможности моделирования творческой деятельности человека непрерывно идут дискуссии и существуют различные точки зрения, как положительные, так и отрицательные. Попытаемся рассмотреть этот вопрос с математической точки зрения. Что есть творчество с точки зрения математика?

Воспользуемся известной теоремой Геделя. Ее смысл состоит в том, что «всякая достаточно мощная формальная непротиворечивая логико-математическая система обязательно содержит формулу, которую в данной системе нельзя ни доказать, ни опровергнуть». Иначе говоря, если имеем систему аксиом $A_1, A_2, \ldots, A_n$, то в этой системе можно сформулировать предложение B_0, которое невозможно ни доказать, ни опровергнуть при помощи данной системы аксиом. Однако может оказаться, что при добавлении к имеющейся системе аксиом некоторой аксиомы A_{n+1} станет возможным доказать или опровергнуть предложение B_0. Но и в этом случае обязательно найдется хотя бы одно еще предложение B_1, которое невозможно ни доказать, ни опровергнуть при помощи уже расширенной системы аксиом. Систему нужно снова расширять и т. д. Так, геометрия Лобачевского содержит в себе геометрию Евклида, а из теории относительности Эйнштейна как частный случай следует ньютоновская механика.

Творчество — это процесс расширения системы, в результате чего невыводимые утверждения становятся выводимыми. Иначе говоря, если некоторая задача не может быть решена в данной логической системе, необходимо искать другую систему, логически более мощную. Тогда творчество заключается в расширении системы, увеличении ее

логической мощи, ее логического «богатства», что дает возможность решения новых задач, не решаемых в старой системе.

Итак, можно дать два определения творчества.

1. Творчество — это поиск инвариантов и соотношений между ними.
2. Творчество — это расширение логической системы с целью решения новых задач.

Так с математической точки зрения можно представить процесс творчества.

Не меньшее методологическое значение для понимания и моделирования процесса творчества имеют теоремы Мак-Каллака и Питтса — основателей направления, называемого нейрокибернетикой. Этими авторами введено понятие математического нейрона. Если нейрон является основной рабочей клеткой коры больших полушарий мозга человека, то математический нейрон есть абстрактный логический элемент, в котором формально отражены лишь те свойства живого нейрона, которые связаны с переработкой информации. Принцип действия математического нейрона и его возможности для решения практических задач изложены в гл. 4. К теме этой главы имеют отношение теоремы Мак-Каллока—Питтса. Смысл их сводится к тому, что любое функционирование живой нервной ткани, которое можно представить с помощью конечного числа слов в терминах логического исчисления высказываний, может быть описано при помощи искусственной нейронной сети. Таким образом, существует принципиальная возможность создания сети из математических нейронов, способной к творческой деятельности.

Впрочем, теоремы Мак-Каллока—Питтса представляют собой *теоремы существования*: они ничего не говорят о том, как нужно создавать сеть из математических нейронов, чтобы воспроизвести творческую деятельность человека, а только утверждают, что такую сеть принципиально можно построить. Практических же успехов в области моделирования творческой деятельности удалось добиться, следуя другим альтернативным направлением искусственного интеллекта, называемым *высокоуровневой стратегией*, или *кибернетикой «черного ящика»*.

8.2. ТВОРЧЕСТВО В МУЗЫКАЛЬНОЙ СФЕРЕ

Человеческий мозг — это своего рода банк данных и знаний, в котором хранится огромный объем информации, собранной за всю прожитую человеком жизнь. Доказано, что человеческий мозг никогда и ничего не забывает. Каждый прожитый им день до мельчайших подробностей, как на видеопленку, записывается в его память, и эта информация может быть определенным образом извлечена. Кроме того, имеются

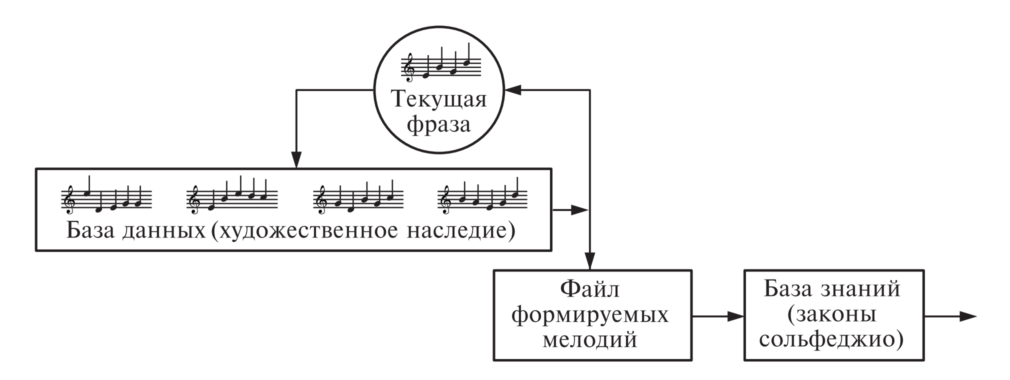


Рис. 8.1. Примерная схема музыкального творчества

попытки доказать, что может быть извлечена информация, переданная человеку от предыдущих поколений и из прожитых ранее жизней.

Точно так же в памяти композитора существует множество различных мелодий, накопленных им в течение жизни, а может быть, даже передавшихся ему с генами от его далеких предков. И естественно полагать, что фрагменты этих мелодий, отдельные музыкальные фразы, музыкальные инварианты осознанно или неосознанно используются композитором в его творческом процессе.

Учитывая все это, первое, что нужно сделать при создании модели музыкального творчества (рис. 8.1), — занести в память компьютера как можно больше музыкальных произведений (создать базу данных). Далее, как и в любой интеллектуальной системе, нужно создать базу знаний, состоящую из законов музыкальной гармонии — соотношений между музыкальными инвариантами (сольфеджио). Теперь на вход компьютера надо подать «творческое вдохновение» в виде музыкальной фразы, состоящей, например, из четырех нот, и заставить компьютер отыскать такое же сочетание нот в одном из хранящихся в его памяти музыкальных произведений. Затем нужно прочитать следующую за найденной фразой ноту и приписать ее к текущей музыкальной фразе. Первую ноту из этой фразы выдать в файл формируемых мелодий и вычеркнуть из текущей фразы так, чтобы в ней по-прежнему оставалось четыре ноты. Процесс поиска следует продолжить, анализируя следующие за найденной в памяти компьютера мелодии.

В результате в файле формируемых мелодий сформируется последовательность нот новой мелодии, которая по своему звучанию будет напоминать заложенные в памяти компьютера известные мелодии, но отличаться от них. Например, если в память закладывались вальсы, то на выходе будет вальс, если марши, то на выходе будет марш и т. п.

После завершения формирования мелодия автоматически аранжируется согласно законам музыкальной гармонии, хранящимся в базе знаний.

Алгоритм выбора продолжений мелодий из базы данных можно снабдить эвристическими правилами, регулируя их силу с помощью все тех же коэффициентов доверия (см. гл. 3). Естественно, что для разных музыкальных стилей будут и разные коэффициенты доверия. Эти коэффициенты можно изменять в процессе обучения, добиваясь улучшения качества сочиняемых мелодий.

Изложенная выше методика компьютерного синтеза музыки лишь в упрощенной форме отражает процесс творчества композитора и не претендует на статус полной модели его деятельности. Тем не менее эта модель в упрощенной форме отражает наши представления о творческой деятельности композитора. В ней есть база данных, содержащая художественное наследие, база знаний, состоящая из законов музыкальной гармонии, а также элемент вдохновения в виде начального приближения, которое можно задавать генератором случайных чисел. Существует также возможность совершенствования таланта компьютерного композитора за счет его обучения, например, путем модификации коэффициентов доверия используемых правил формирования продолжений мелодии.

Интенсивное применение методов искусственного интеллекта в музыке началось в середине XX в. В 1957 г. американские исследователи М. Мэтьюз и Н. Гутман положили начало развитию целого класса музыкальных языков программирования: MUSIC, C-Sound и т. д. Проведенные ими эксперименты показали, что ряд искусственно сочиненных музыкальных произведений удовлетворяла критерию Тьюринга: слушатели-эксперты не смогли отличить машинную музыку от авторской.

Методики, отражающие другие стороны музыкального творчества, были предложены в 1955 г. учеными Иллинойского университета Л. Хиллером и Л. Исааксом. Они провели серию экспериментов, в которых последовательно закладывались законы сольфеджио в базу знаний (гармонизация) и вводились разнообразные музыкальные ритмы и темпы. Был также использован датчик случайных чисел.

На рисунке 8.2 изображена блок-схема программы, реализованной на машине «Иллиак». На входе программы генератором случайных чисел задавались целые числа, при помощи которых были закодированы нотные знаки. Каждое число пропускалось через последовательность из четырех контрольных схем (I—IV). Эти схемы пропускали в запомина-

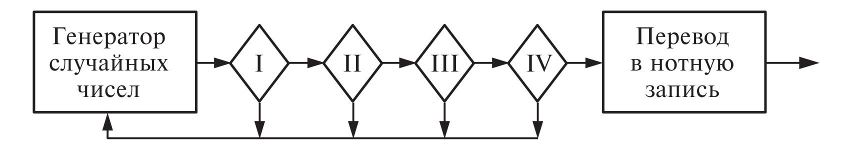


Рис. 8.2. Блок-схема программы компьютерного сочинения музыки

ющее устройство только те числа, которые образовывали правильную (подчиняющуюся заложенным в контрольной схеме ограничениям) мелодическую линию. Законченный период запоминался, а затем выдавался на печать и расшифровывался в виде нот. Если же хотя бы одна из схем задерживала хотя бы одну ноту, то управление вновь передавалось генератору случайных чисел и поиски правильной ноты продолжались. После 50 неудачных попыток подобрать нужную ноту мелодическая линия разрушалась и начинала выстраиваться новая линия. За 1 ч работы машина «Иллиак» создавала 100 мелодий. Именно так была написана знаменитая «Иллиак-сюита» для струнного оркестра.

Широкую известность в свое время получили музыкальные произведения, сочиненные ЭВМ «Урал-2» по алгоритмам, разработанным советским математиком Р. Х. Зариповым. В его программах также использовался генератор случайных чисел, который выдавал не только код ноты, но и длительность и интервал между нотами. Эти данные проходили контроль на соответствие закономерностям базы знаний — законам музыкальной гармонии, полученным при анализе широкого круга музыкальных произведений. Программы Р. Х. Зарипова моделировали сочинение мелодий в мажоре или миноре, аккомпанемента к ним, сочинение мелодий на заданный стихотворный текст и ритм.

Принципы, разработанные первыми программистами-музыковедами, в настоящее время закладываются в схемы современных оркестровых электромузыкальных инструментов и широко используются композиторами и музыкантами. Однако этот инструментарий является вспомогательным, так как его применение ограничено написанием гармонии, аранжировок, сопровождений. Сами же темы по-прежнему сочиняет человек. Дело в том, что создать хорошую простую мелодию неизмеримо сложнее, чем оркестровое произведение в авангардистской манере, перегруженное случайными звукосочетаниями и диссонансами. Когда композитор сочиняет мелодию, которая становится популярной, происходит колоссальный прорыв вперед, так как это открытие нового, неизвестного ранее соотношения между музыкальными инвариантами.

Современные компьютерные программы могут сочинять новые мелодии, которые приятно звучат и чем-то напоминают ранее известные, имеют хороший стиль и манеру, но в них всегда чего-то не хватает, чтобы стать действительно популярными. Знаменитые «Yesterday» и «Лунная соната», да и не только они, сочинены не компьютером! Однако очень вероятно, что компьютер превзойдет человека и в этой области. Как это ни обидно сознавать, возможно, уже в совсем недалеком будущем молодежь будет петь и танцевать под компьютерную музыку, а великие произведения, сочиненные классиками, будут считаться экзотикой…

8.3. ТВОРЧЕСТВО В ПОЭЗИИ

Считается, что задача моделирования стихотворчества несоизмеримо сложнее, чем задача моделирования сочинения музыкальных произведений. Как показали исследования русского языка, если одна буква делового языка несет 0,6 бит информации, буква обыкновенной разговорной речи — 1 бит, то буква поэтической речи — 1,5 бит.

Не вдаваясь в философские и этические проблемы, подойдем к проблеме моделирования поэтического творчества с помощью известных инструментальных средств. Например, можно воспользоваться известным *формализмом Бэкуса—Наура*. Этот формализм применяется главным образом в связи с распознаванием и обработкой текстовой информации, машинным переводом, а также созданием систем естественно-языкового общения. В формализме применяются следующие символы-операторы:

::= — «определяется как» или «может быть переписан как»;

| (вертикальная черта) — используется для разделения различных альтернативных возможностей;

⟨⟩ (угловые скобки) — используются для заключения нетерминальных символов, т. е. символов, которые должны определяться одним из правил. В отличие от них символы, не заключенные в угловые скобки, считаются терминальными и представляют лишь самих себя.

Например, цифра в формализме Бэкуса—Наура определяется следующим образом:

$$\langle \text{цифра} \rangle ::= 0|1|2|3|4|5|6|7|8|9,$$

где ⟨цифра⟩ — нетерминальный символ, а каждая из цифр 0, 1, …, 9 — терминальный символ.

Приведем некоторое множество правил упрощенного описания синтаксической структуры английского предложения.

1. $\langle \text{предложение} \rangle ::= \left\langle \begin{matrix}\text{существ.}\\ \text{фрагмент}\end{matrix} \right\rangle \left\langle \begin{matrix}\text{глагольн.}\\ \text{фрагмент}\end{matrix} \right\rangle \quad \left\langle \begin{matrix}\text{существ.}\\ \text{фрагмент}\end{matrix} \right\rangle$
2. $\left\langle \begin{matrix}\text{существ.}\\ \text{фрагмент}\end{matrix} \right\rangle ::= \langle \text{артикль} \rangle \left\langle \begin{matrix}\text{определяемое}\\ \text{существ.}\end{matrix} \right\rangle$
3. $\left\langle \begin{matrix}\text{определяемое}\\ \text{существ.}\end{matrix} \right\rangle ::= \langle \text{существ.} \rangle | \langle \text{определение} \rangle \left\langle \begin{matrix}\text{определяемое}\\ \text{существ.}\end{matrix} \right\rangle$
4. $\left\langle \begin{matrix}\text{глагольн.}\\ \text{фрагмент}\end{matrix} \right\rangle ::= \langle \text{глагол} \rangle | \left\langle \begin{matrix}\text{глагольн.}\\ \text{фрагмент}\end{matrix} \right\rangle | \langle \text{фрагмент наречия} \rangle$
5. $\left\langle \begin{matrix}\text{фрагмент}\\ \text{наречия}\end{matrix} \right\rangle ::= \langle \text{наречие} \rangle | \langle \text{предлог} \rangle \left\langle \begin{matrix}\text{существ.}\\ \text{фрагмент}\end{matrix} \right\rangle$

К этим правилам мы добавим небольшой словарь.

1. ⟨артикль⟩ ::= a|the

2. ⟨существ.⟩ ::= cat|mouse|dog|fox (кошка, мышь, собака, лиса)
3. ⟨определение⟩ ::= quick|lazy|brown|*black* (быстрый, ленивый, бурый, черный)
4. ⟨глагол⟩ ::= goes|jumps|runs (идет, прыгает, бежит)
5. ⟨наречие⟩ ::= quickly|easily (быстро, легко)
6. ⟨предлог⟩ ::= over|under|through (над, под, через)

Приведенное множество синтаксических правил обычно используется для синтаксического анализа текстовой информации. Однако этот же фрагмент можно использовать вместе с генератором случайных чисел для создания случайных предложений, подчиняющихся указанным синтаксическим правилам. Иными словами, можно запустить процедуру синтаксического анализа как бы в обратную сторону.

Чтобы построить какое-нибудь предложение, процесс генерации нужно запустить с правила 1. Здесь имеются две возможности, и для выбора какой-то одной из них следует обратиться к генератору псевдослучайных чисел. Затем процесс генерации развивается в соответствии с выбранной альтернативой. Если выбрана первая из возможностей, то управление сначала должно перейти к правилу 2 для создания «существ. фрагмента», а после того, как это сделано, — к правилу 4 для создания «глагольн. фрагмента». Поскольку правила 2 и 4 в свою очередь передают управление другим правилам, то необходимо иметь *стек* («магазинную память») для напоминания о неоконченных моментах в различных правилах. Когда осуществляется первая передача управления от правила 1 к правилу 2, в стек помещаются первый элемент и указатель на следующий по порядку элемент (в соответствии с выбранной альтернативой).

Если в процессе генерации встречается терминальный символ, то он добавляется к выходной строке, которая создается.

Приведенный способ генерации предложений позволяет получить такие фразы, как: «The quick brown fox jumps over the lazy dog» — «проворная бурая лиса перепрыгнула через ленивую собаку».

Всего в словаре приведенного множества содержится 18 слов, выступающих в качестве терминальных символов: два артикля, четыре существительных, четыре прилагательных и т. д. Однако приведенный метод применим и к большим словарям, содержащим яркие, выразительные слова. Таким образом, можно заставить компьютер сочинять стихи, в которых, однако, не будет соблюдаться рифма. Для выдерживания рифмы необходимо введение дополнительных правил, рифмующих фразы.

Для того чтобы сочиненные компьютером произведения имели смысл, необходимо ввести базу соответствующих знаний. Каждое вводимое в словарь слово должно быть увязано с другими не только синтаксическими, но и семантическими связями. Различные сочетания слов должны быть оценены некоторыми оценивающими параметра-

ми, задающими уровень смыслового соответствия. Такие параметры должны использоваться при построении фраз, подобно тому как в экспертных системах при получении заключений используются коэффициенты доверия. Этими коэффициентами можно регулировать уровень осмысленности и степень абстрагизма создаваемого произведения, определять его характер, жанр и смысловую направленность.

Как и в других интеллектуальных системах, в системах стихотворчества коэффициенты доверия могут меняться в процессе работы программы, т. е. алгоритмы стихотворчества могут быть обучаемыми.

Таким образом, существующий уровень развития инструментальных средств и методов искусственного интеллекта позволяет создать более или менее приемлемые алгоритмы поэтического творчества. Это свидетельствует о том, что и этот вид человеческой деятельности в принципе поддается компьютерному моделированию.

8.4. ТВОРЧЕСТВО В НАУКЕ

Имитация творческой деятельности ученых — одна из главных задач научной области «искусственный интеллект». И на этом пути достигнуто немало впечатляющих результатов. Достаточно упомянуть: первую экспертную систему Р. Луллия, ставящую диагнозы и делающую прогнозы на основании имеющихся научных знаний; разработанную А. Ньюэллом и Г. Саймоном знаменитую программу «Логик-теоретик», автоматически доказывающую и формулирующую новые теоремы алгебры логики; многочисленные применения нейронных сетей, позволившие выявить и использовать неизвестные ранее закономерности исследуемых предметных областей.

В этом разделе мы остановимся только на одной, но актуальной для XXI в. научной проблеме, которая пока еще ждет своего творческого решения. Речь пойдет о *современном кризисе прикладной математики*.

Развитие нашей цивилизации свидетельствует о продолжающемся повышении роли математического компьютерного моделирования как в научных исследованиях, так и в различных областях практической деятельности человека. Одним из наиболее популярных инструментов математического компьютерного моделирования является формулировка и решение краевых задач математической физики.

В истории развития методов решения краевых задач можно проследить три периода. Первый исторический период, продлившийся примерно до середины XX в., начался с основополагающих работ Ж. Л. Д'Аламбера и Ж. Б. Ж. Фурье, выполненных в XVIII — начале XIX в. Путем разделения переменных им удалось получить ряд решений дифференциальных уравнений в частных производных для

простейших областей, называемых каноническими, — круга, квадрата, цилиндра, шара и пр. Затем, на протяжении последующих полутора веков, усилия математиков в этой области в основном сводились к развитию метода разделения переменных и изобретению других приемов, позволяющих получить решение той или иной краевой задачи для других дифференциальных уравнений, для других областей, с другими краевыми условиями. Каждое такое решение было событием в математическом мире и отмечалось присуждением премий и присвоением регалий. Метод математического моделирования был доступен узкому кругу математиков-профессионалов, деятельность которых представляла собой *творческий процесс сродни деятельности поэтов, художников, композиторов*.

Появление в середине XX в. быстродействующих электронно-вычислительных машин изменило эту ситуацию. Оказалось, что если разбить область решения краевой задачи на множество мелких подобластей (рис. 8.3) и для каждой подобласти ввести гипотезы, упрощающие физические свойства среды, то процесс интегрирования дифференциальных уравнений можно свести к множеству элементарных арифметических действий. Таким образом, краевые задачи математической физики стало возможным решать с помощью ЭВМ «с позиции грубой силы», получая решение не в виде аналитических формул, а в виде массивов чисел.

Так появилась на свет новая отрасль математики, называемая *дискретной*. На смену классическим аналитическим методам пришли численные алгоритмы, с помощью которых удалось создать универсальные пакеты прикладных программ, оснащенных удобными сервисными средствами. Математическое компьютерное моделирование стало общедоступным и из творчества превратилось в ремесло. Математики-аналитики с их хитроумными математическими выкладками, казалось, навсегда утратили свой авторитет и отошли в прошлое.

Однако, как утверждают философы, жизнь развивается по спирали. Маятник, качнувшийся в одну сторону, должен обязательно отклониться и в другую. Увлечение численными методами в полной мере выявило не только их бесспорные преимущества, но и неустранимые недостатки. К числу последних относится невозможность надежной оценки погрешности расчетных результатов. Этот недостаток особенно ощутим в последнее время в связи с применением метода математического моделирования для расчета ответственных объектов и процессов, от которых зависит безопасность людей, государств, цивилизации.

Следует заметить, что математический аппарат, которым пользовались математики минувших веков, был более надежен. Дело в том, что решения, полученные аналитическими методами в виде аналитических формул, могут быть проверены на удовлетворение дифференциальным уравнениям и краевым условиям решаемой задачи, т. е. их погрешность может быть оценена.

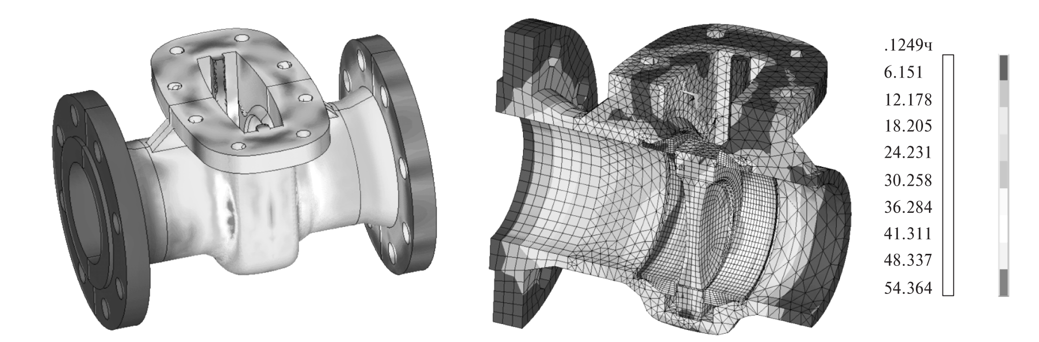


Рис. 8.3. Расчетная область — промышленный объект, разбита на множество подобластей — конечных элементов. Решение краевой задачи представлено в виде цветной картинки — каждому цвету соответствует свой уровень напряжений

Решения же, получаемые численными методами, представляют собой массивы чисел, о погрешности которых судят только по тому, как эти числа изменяются с увеличением количества разбиений расчетной области. Обычно полагают, что результатам можно доверять, если они перестают изменяться с измельчением сетки. Однако уже давно показана теоретическая несостоятельность этого подхода [12, 66, 68, 86, 87]. Дело в том, что с измельчением конечноэлементной сетки ухудшается обусловленность матриц систем разрешающих алгебраических уравнений. Это значит, что с измельчением сетки численные решения сходятся совсем не к искомым решениям краевых задач, что делает практически невозможной надежную оценку точности расчетных результатов.

К результатам, полученным численными методами, следует относиться крайне осторожно, особенно если речь идет о расчетах объектов и процессов ответственного назначения. Тем не менее на современном рынке программных средств имеется множество компьютерных программ, реализующих численные методы решения краевых задач теплопроводности, гидродинамики, теории упругости, теории электрических, магнитных, гравитационных и даже торсионных полей. Эти пакеты снабжены удобными сервисными средствами, так что любой инженер может получать численные решения краевых задач практически любой степени сложности. Но возникла другая серьезная проблема: оценить погрешности таких численных решений для сложных инженерных задач, как правило, не представляется возможным. В книге Л. Н. Ясницкого и Т. В. Данилевич «Современные проблемы науки» [68], а также в работах [66, 86, 87] эта парадоксальная ситуация названа «*современным кризисом прикладной математики*».

По прогнозам специалистов, XXI век — это век жестоких техногенных катастроф, стихийных и экологических бедствий. Все чаще приходится слышать сообщения о падении ракет, самолетов, взрывах на ядерных станциях и промышленных объектах, обрушениях зданий. Среди прочих причин трагедий называются ошибки проектирования. Яркий пример — обрушение Московского аквапарка. Показательно то, что здесь явно прослеживаются попытки сокрытия истинных причин обрушения. В первоначальных заключениях следственных комиссий обнаруживались и следы терроризма, и плохое качество цемента, и нехватка колонн для поддержания крыши. И только после обрушения второго здания, спроектированного тем же авторским коллективом, — здания Басманного рынка, причины трагедии стали очевидными для всех. Последнюю черту под расследованиями подвело телевизионное заявление руководителя проектов Надара Канчелли: «... во всем виновата компьютерная программа ...», с помощью которой выполнялись прочностные расчеты.

Если причины обрушения современных зданий, в конце концов, как-то выясняются, то гораздо хуже обстоят дела с расследованиями причин авиационных катастроф. Перед членами следственных комиссий, как правило, работниками заводов, создавших рухнувшие самолеты, ставится задача — «не запятнать честь своих коллективов!». В результате, по данным Межгосударственного авиационного комитета, в более чем 80% случаев виновниками авиакатастроф признаются пилоты (которые погибли и потому не могут оправдаться). Получается, что более 80% пилотов всех разбившихся самолетов были террористами или у них у всех без исключения суицид. Статистические данные Межгосударственного авиационного комитета явно противоречат здравому смыслу.

Как показано в работах [12, 66, 68, 86, 87], есть все основания полагать, что одной из причин наблюдающегося в последнее время роста техногенных аварий и катастроф является низкое качество современных компьютерных программ, используемых инженерами при проектировании конструкций ответственного назначения. Есть основания ожидать, что современный кризис прикладной математики перерастет в кризис современной цивилизации.

Сейчас, как никогда прежде, стали актуальными вопросы точности компьютерного математического моделирования. Малейшие ошибки в математических моделях, в частности в методах решения краевых задач, приводят к тяжелым экологическим, экономическим и социальным последствиям. Пришло время применять только такие модели и методы, которые могут гарантировать необходимую надежность расчетных результатов.

Выход из кризиса следовало бы искать в том, чтобы вообще отказаться от численных методов решения краевых задач и применять только те методы, которые приводят к точным аналитическим решени-

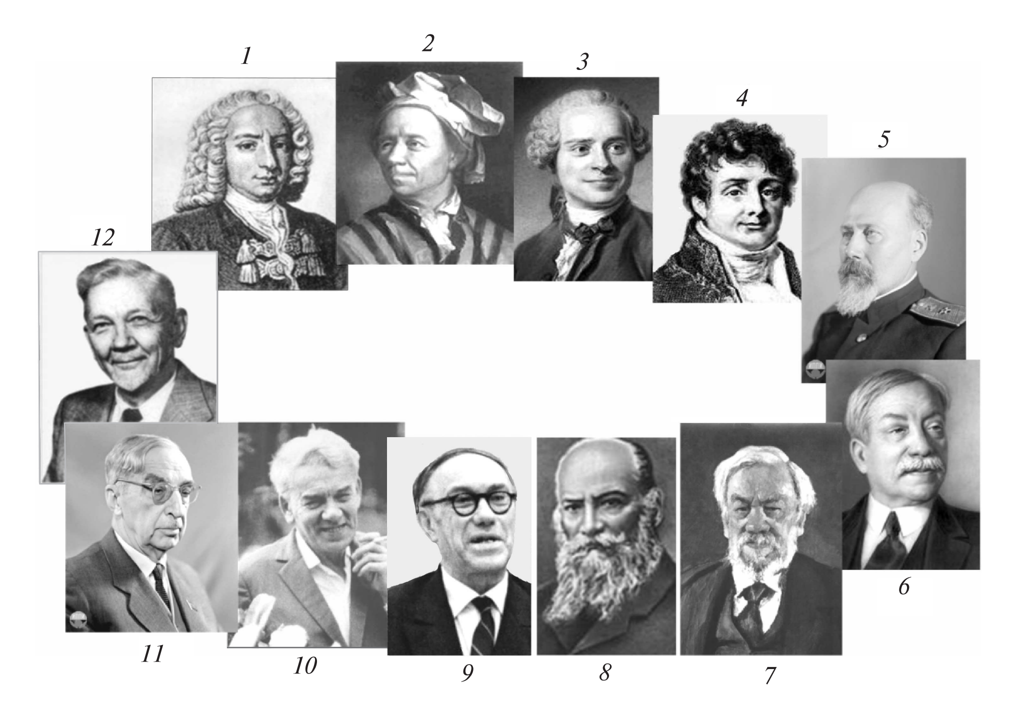


Рис. 8.4. Математики-профессионалы XVIII, XIX, XX вв. – авторы точных решений краевых задач: *1* — Даниил Бернулли; *2* — Леонард Эйлер; *3* — Жан Лерон Даламбер; *4* — Жан Батист Жозеф Фурье; *5* — Петр Федорович Папкович; *6* — Борис Григорьевич Галеркин; *7* — Сергей Александрович Чаплыгин; *8* — Николай Егорович Жуковский; *9* — Михаил Александрович Лаврентьев; *10* — Анатолий Николаевич Лурье; *11* — Николай Иванович Мусхелишвили; *12* — Степан Прокофьевич Тимошенко

ям. Но на современном рынке программной продукции программных пакетов, реализующих такие методы, практически нет. Их нет потому, что аналитические методы не универсальны и крайне плохо поддаются алгоритмизации. Они требуют *творческого*, интеллектуального подхода к решению практически каждой новой краевой задачи. Это связано с длительной и кропотливой работой высокообразованных математиков, с привлечением их *опыта, интуиции, таланта*. Точные решения краевых задач могли получать математики-аналитики XVIII–XIX вв. (рис. 8.4), причем только для простейших областей. Сейчас же традиции научной школы математиков-аналитиков, создававшейся на протяжении нескольких последних веков, к сожалению, в значительной степени утрачены.

Более правильной представляется стратегия выхода из кризиса, состоящая в том, чтобы воспользоваться результатами очередной ком-

пьютерной революции — успехами в области искусственного интеллекта. Надо научиться моделировать интеллект математиков-аналитиков прошлых веков, их интуицию, опыт, талант. Идея состоит в том, чтобы, моделируя интеллект математиков-профессионалов, научить компьютер получать точные аналитические решения любых краевых задач, таких, которые необходимы современным инженерам.

Первые попытки компьютерной имитации творческой деятельности математика-аналитика показали перспективность нового подхода. На сайте Пермского отделения Научного совета РАН по методологии искусственного интеллекта **www.PermAI.ru** в разделе «Проекты» размещен доступный для общего пользования программный пакет «REGIONS». Этот пакет, по существу, является экспертной системой, которая имитирует творческую деятельность математика-профессионала (эксперта), выполняющего решение краевой задачи. В основу пакета положен аналитический метод решения краевых задач — метод фиктивных канонических областей (ФКО). Этот метод, предложенный еще в начале 1970-х гг. Л. Н. Ясницким [77, 81], развитый В. А. Елтышевым [20] и С. Л. Гладким [12], хотя и позволил в свое время найти точные аналитические решения ряда практически важных инженерных задач [12, 63, 77, 80, 85], однако не нашел широкого применения ввиду отсутствия универсальности. Дело в том, что успех решения краевой задачи методом ФКО в сильной степени зависит от опыта и интуиции применяющего его математика. Теперь же появилась возможность заложить в компьютер интеллект математиков-профессионалов в виде набора эвристических правил. Эти правила в большинстве случаев не имеют строгих математических доказательств и поэтому обладают различными коэффициентами доверия. Мы имеем дело с ***нечеткой математикой***, характерной для систем искусственного интеллекта. Результат же применения нового подхода — точное аналитическое решение краевой задачи, не нуждающееся в оценке погрешности и не вызывающее сомнений в своей надежности.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите общие принципы построения математических моделей.
2. Дайте определение инварианта и приведите примеры инвариантов, известных вам из математики, физики, искусства.
3. Сформулируйте теорему Геделя и поясните, какое отношение она имеет к творчеству.
4. Что такое творчество с точки зрения теории инвариантов?
5. Сформулируйте теорему Мак-Каллока — Питтса и поясните, какое отношение она имеет к творчеству.

6. Если бы вам предстояло писать программу компьютерного сочинения музыки, какую бы блок-схему вы предложили?
7. Поясните, каким образом можно использовать для сочинения стихов формализм Бэкуса—Наура. Как можно с помощью этого алгоритма придавать сочиненным произведениям смысл, изменять степень абстракции, определять его характер, жанр?
8. В чем состоит и как проявляется современный кризис прикладной математики? Возможно ли его преодолеть методами искусственного интеллекта?

Глава 9

НАСТОЯЩЕЕ И БУДУЩЕЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

9.1. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

По мнению многих исследователей, единого общепризнанного определения искусственного интеллекта в настоящее время не существует, как не существует и универсального определения человеческого интеллекта.

В ранний период возникновения и становления науки об искусственном интеллекте с этим понятием связывали надежды на создание машин, способных мыслить и чувствовать, как человек, способных соперничать с человеческим мозгом и превзойти его. Эти идеи, на долгое время захватившие воображение энтузиастов, вызвали волну споров и критики. Они поставили ряд глобальных философских вопросов, так и оставшихся без ответа:

- Можно ли создать мыслящую машину, если мы сами не знаем до конца, что такое мышление и каковы его механизмы?
- Как мы можем понять и оценить, что у машины есть сознание, чувства, эмоции?
- Не деградирует ли человечество, не превратится ли оно в слуг создаваемого им искусственного разума?

В результате в искусственном интеллекте, как и во множестве других научных областях, произошло разделение исследователей на «романтиков» — теоретиков, занимающихся глобальными вопросами создания искусственного разума, и «прагматиков» — практиков, развивающих и реализующих идеи искусственного интеллекта для решения локальных практических проблем. Первое направление иногда называют «*сильным искусственным интеллектом*» (Strong AI), а второе — «*слабым искусственным интеллектом*» (Weak AI). Чтобы отделить себя от «философов», прагматики стали называть себя разработчиками *интеллектуальных систем* или *систем искусственного интеллекта*.

Обе эти ветви искусственного интеллекта сосуществуют, взаимно дополняя и обогащая друг друга. Так, фундаментальная идея создания глобального искусственного разума в виде универсальной формулы знаний о мире, высказанная более 700 лет назад Раймундом Луллием (см. § 1.1), нашла блестящее воплощение в экспертных системах (гл. 3), впервые превратив искусственный интеллект в высокорентабельную сферу бизнеса. Идеи Мак-Каллока—Питтса—Розенблатта (см. §§ 4.1–4.5), предложивших способ моделирования мозга на

структурном уровне, стали основой *нейроинформатики* — науки, побившей все рекорды чрезвычайным многообразием практических приложений. Попытки объяснить механизм возникновения разума как результат взаимодействия и развития совокупности относительно простых сущностей, предпринятые Гербертом Саймоном, Джоном Холландом и другими исследователями, привели к появлению эффективных инструментов решения широкого круга прикладных задач — генетическим алгоритмам и мультиагентным системам (§ 6.4).

В то же время, успехи практического применения интеллектуальных систем являются, во-первых, критерием правильности заложенных в их основу гипотез и, во-вторых, стимулом дальнейшего развития и совершенствования этих гипотез, поводом для исследований мозга, раскрытия его новых тайн, все более полного понимания человеческого интеллекта.

По-видимому, человеческий мозг так же, как и сам человек, никогда не будет познан до конца. Люди будут постоянно решать проблемы сильного искусственного интеллекта, сталкиваясь со все новыми трудностями и преодолевая их. И этот процесс видится бесконечным. Но тогда бесконечен и другой процесс — процесс создания и развития все более и более совершенных интеллектуальных систем.

9.2. НЕЙРОННЫЕ СЕТИ И ЭКСПЕРТНЫЕ СИСТЕМЫ

По мнению многих исследователей, интеллект — это очень сложная область знаний, которую невозможно описать с помощью какой-то одной теории. Поэтому ученые строят целую иерархию теорий, характеризующих интеллект на разных уровнях абстракции. На самом нижнем уровне этой иерархии находятся нейронные сети, генетические алгоритмы и другие формы эволюционирующих вычислений, а на более высоком уровне абстракции — экспертные системы.

Остановимся на двух крайних иерархических ступенях — самой верхней и самой нижней. Отметим для начала, что строительство «здания» искусственного интеллекта началось не снизу, а сверху, причем с огромным разрывом во времени. Идея создания интеллектуальной системы, основанной на явных знаниях, была высказана более 700 лет назад Раймундом Луллием, а идея математического нейрона была опубликована Мак-Каллоком и Питтсем в 1943 г. Своего апогея в области практических приложений технология экспертных систем достигла в середине 80-х гг. XX в. Именно тогда, с конца 1980-х гг., и началась конкуренция на рынке программных приложений между экспертными системами и нейронными сетями, которая продолжается и по настоящее время.

Строя прогнозы на будущее, многие авторы отмечают наметившуюся тенденцию к сближению этих двух крайностей в виде появления

гибридных интеллектуальных систем, использующих сразу обе технологии. Тем не менее очень часто при решении практических задач приходится выбирать, какой из этих двух конкурирующих технологий отдать предпочтение, поэтому укажем их плюсы и минусы.

Основным *недостатком нейронных сетей* является их невербальность. Если вывод, полученный экспертной системой, всегда можно логически обосновать, запустив «подсистему объяснений» (см. § 3.2), то в случае применения нейронной сети такая возможность отсутствует. Другой их недостаток — отсутствие законченной фундаментальной теории и, как следствие, отсутствие надежных и достаточно универсальных методик проектирования и обучения нейросетей, гарантирующих успех их применения. По этому поводу программисты шутят, что нейросетевые технологии все еще не стали в полной мере наукой и содержат элементы искусства и даже религии.

Одним из главных *преимуществ нейросетевых технологий* является сравнительно низкая трудоемкость их создания и, как следствие, низкая себестоимость программной продукции. При нейросетевом моделировании отсутствуют такие трудоемкие и дорогостоящие процедуры, как извлечение знаний и их формализация (см. § 3.3—3.5). Вместо этого предлагается целый арсенал алгоритмов обучения, осуществляющих автоматическое извлечение знаний, и стандартный способ их хранения в виде сил синаптических связей.

Нейросети дают стандартный способ решения многих нестандартных задач. Если для создания средней по сложности экспертной системы требуются месяцы и годы работы коллектива специалистов, включающего инженера-когнитолога, программиста, эксперта и пользователя (см. § 3.2—3.3), то вполне прилично работающая нейронная сеть может быть сгенерирована всего с нескольких попыток и обучена рядовым пользователем нейропакета, скачанного из сети Интернет.

К серьезному преимуществу нейросетевых технологий можно отнести принципиальную возможность выявления новых, еще неизвестных экспертам знаний. Нейросети особенно эффективны там, где нужен аналог человеческой интуиции, где законы и закономерности предметной области неизвестны, где существуют проблемы с их выявлением и формализацией.

В противоположность нейросетям, *экспертные системы* оперируют только с теми знаниями, законами и закономерностями предметных областей, которые заложены в нее экспертом. В некоторых случаях экспертные системы тоже могут образовывать неизвестные экспертам логические цепочки и, таким образом, генерировать новые знания. Но эти знания формируются на основе правил, имеющихся в базе знаний, а потому, согласно терминологии теоремы Геделя (см. § 8.1), «не расширяют аксиоматическую систему». Следовательно, процесс их получения не является актом творчества. Другими словами, знания, которые генерируют экспертные системы, не могут быть

принципиально новыми. Они не выходят за рамки знаний экспертов — авторов экспертной системы. В этом состоит их недостаток по сравнению с нейронными сетями.

Как убедительно показывает мировой опыт, в частности материал данной книги, хорошо спроектированные и правильно обученные нейронные сети способны самостоятельно выявлять закономерности практически любых предметных областей и строить адекватные математические модели в промышленности, экономике и бизнесе, политологии, криминалистике, медицине, экологии, исторических науках и др. Причем, многократно фиксировались случаи, когда в процессе моделирования нейросети выявляли и использовали новые, неизвестные и даже необъяснимые пока знания и закономерности [79].

Так, в работе [71] (см. рис. 5.5) была спрогнозирована победа Д. А. Медведева за полтора года до президентских выборов 2008-го года, когда его личность как политика еще была мало известна. В работах [82] и [89], опубликованных в 2008 и 2010 гг., когда президент Д. А. Медведев был на вершине своей популярности, ему прогнозировалось снижение рейтинга (см. рис. 5.6, *а*), тогда как другому политику — В. В. Жириновскому — прогнозные кривые предсказали постепенный рост популярности (см. рис. 5.6, *б*), что и наблюдалось в последующие годы.

Другой пример — в результате разработки нейросетевой системы диагностики заболеваний сердечно-сосудистой системы выявлены новые, неисследованные ранее медицинские знания и закономерности. В результате чего сделан вывод о необходимости корректировки существующей практики профилактики и лечения кардиологических заболеваний [69, 109] (см. § 5.2).

При разработке нейронной сети, предназначенной для определения способности человека к предпринимательской деятельности [84], (см. § 5.7) также выявлено влияние нетипичных факторов (например, цвет глаз). Учет этих факторов позволил повысить точность прогнозов, что, однако, не поддается логическому объяснению в рамках известных знаний.

В нашей практике [79] было много случаев, когда с помощью нейронных сетей удавалось выявлять неизвестные ранее закономерности, причем такие, которые в рамках традиционной науки объяснить не удавалось, либо объяснения находились, но спустя некоторое время, после длительных обсуждений и после более детального изучения и экспериментального подтверждения.

Вместе с тем не следует переоценивать возможности нейросетевых технологий. Обычно нейросети уступают детерминированным моделям в тех предметных областях, в которых знания известны и могут быть хорошо описаны и формализованы. Нейронные сети нет смысла применять и там, где выходные параметры не зависят или слабо зависят от входных параметров.

Опыт показывает, что нейросети целесообразно использовать при следующих условиях, если:

1) решение задачи затруднительно для человека;
2) при решении задачи можно выделить множество входных факторов (сигналов, признаков, данных и т. п.) и множество выходных факторов;
3) изменения входных факторов приводит к изменению выходных факторов;
4) отсутствует возможность построения детерминированной математической модели предметной области, основанной на явных знаниях.

Несмотря на значительное количество уже известных практических приложений искусственных нейронных сетей, возможности их дальнейшего использования для обработки сигналов окончательно не исчерпаны. Можно предположить, что искусственные нейронные сети еще в течение многих лет будут одним из основных инструментов поддержки принятия решений в условиях отсутствия точных моделей реальных процессов и явлений.

9.3. НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ И МЕТОДЫ РЕГРЕССИОННОГО АНАЛИЗА

Как известно, нейросетевые и нейрокомпьютерые технологии появились в середине XX в. как новая стратегия искусственного интеллекта. Если до их появления искусственный мозг создавался как «черный ящик» (т. е. не важно, что внутри, лишь бы он решал интеллектуальные задачи), то появление нейрокомпьютеров и нейронных сетей ознаменовало начало новой эры — создание компьютерных программ и компьютеров «по образу и подобию» человеческого мозга. Нейротехнологии стали называть низкоуровневой стратегией искусственного интеллекта, подчеркивая, что речь идет о моделировании на самом нижнем уровне абстрагирования — на уровне кирпичиков-нейронов, из которых состоит мозг.

Однако математики обратили внимание на то, что за интригующими словами о моделировании мозга кроется в общем-то несложный математический аппарат, ориентированный на решение традиционной математической задачи аппроксимации данных. Действительно, с математической точки зрения обученная нейронная сеть — это нелинейная векторная функция, аппроксимирующая статистические данные. Она связывает входные (управляющие) и выходные (управляемые) случайные величины. Такие функциональные зависимости в математике принято называть регрессионными уравнениями, поэтому нейросетевые технологии можно классифицировать как один из новых методов построения регрессионных уравнений.

Чем этот новый метод отличается от классических методов регрессионного анализа? Чтобы ответить на этот вопрос, заметим прежде, что классические методы регрессионного анализа предполагают построение регрессионной модели, как любой линейной или нелинейной функции [104], без каких-либо ограничений на их вид. При этом математические формулы не связываются с какими-либо физическими или иными интерпретациями.

В противоположность классическим технологиям регрессионного анализа, нейрокомпьютерые и нейросетевые технологии имеют глубокий физический, психологический и общефилософский смысл. Они моделируют мозг, причем как на функциональном, так и на структурном уровнях.

Методы классического регрессионного анализа преследуют цель аппроксимации статистических данных любыми математическими функциями, в большинстве случаев — линейными. Нейронные сети и нейрокомпьютеры тоже строят аппроксимации, но в этих аппроксимациях используются только такие математические функции, согласно которым, по мнению биологов, функционируют нейроны мозга. Преобразования данных с помощью этих математических функций выполняются в той же последовательности, какая обнаружена биологами в натурных нейроэкспериментах.

Если методы классического регрессионного анализа реализуют идеи гениальных людей-математиков, то нейрокомпьютерные и нейросетевые технологии реализуют изобретение самой Природы (Бога) — принципы построения и функционирования мозга. Следствием этих принципиальных различий являются различия возможностей нейросетевого и регрессионного моделирования.

Нейросетевые технологии обладают следующими преимуществами по сравнению с традиционными методами регрессионного анализа:

- Применение классических методов регрессионного анализа требует введения гипотез о характере причинно-следственных связей между входными и выходными переменными, тогда как в случае применения нейросетевых технологий этого не требуется.
- Применение классических методов регрессионного анализа не допускает наличия линейной корреляции между входными параметрами; для нейросетевых технологий данное требование соблюдать не обязательно.
- При применении классических методов регрессионного анализа часто приходится кластеризовать и без того немногочисленные наблюдения; нейросетевые технологии в большинстве случаев позволяют обойтись без предварительной кластеризации данных.
- Для построения качественных нейросетевых моделей обычно требуются существенно меньшие объемы статистических данных, чем для построения регрессионных моделей.

- Статистические методы хорошо развиты только для одномерных случайных величин. Многомерные статистические модели либо предполагают гауссово распределение наблюдений, что часто не выполняется на практике, либо не обоснованы теоретически. В многомерной статистике, за неимением лучшего, нередко применяют малообоснованные эвристические методы, которые по своей сути очень близки к технологии нейронных сетей.
- При применении нейросетевых технологий имеется возможность распараллеливания вычислительных процессов между отдельными нейронами, в результате чего получается значительный выигрыш по времени.
- Встречаются случаи, когда применение нейросетевых технологий дает положительный результат, в то время как методы классического регрессионного анализа оказываются бессильны.

В качестве недостатка метода нейросетевого моделирования по сравнению с классическими методами регрессионного анализа можно указать недостаточную развитость теоретической базы. Из этого вытекает отсутствие четких инструкций по разработке оптимальных нейросетевых моделей, учитывающих особенности конкретных предметных областей. Именно по этому поводу некоторые специалисты называют процесс создания нейросетевых моделей в большей степени искусством, чем наукой.

Преимуществом классических регрессионных моделей является также возможность объяснения моделируемых явлений. Так, если входные данные имеют одинаковую размерность и нормированы, то иногда по величине коэффициентов перед слагаемыми регрессионного уравнения судят о величине статистической значимости вклада соответствующего входного параметра в результат моделирования, т. е. судят о его объясняющих качествах. В нейронных же сетях подобные оценки обычно затруднены из-за сложности их структуры. С целью упрощения нейронных структур и их вербализации авторы [14] рекомендуют способ редуцирования, после применения которого нейросеть становится прозрачной для понимания. Однако применение этого метода приводит к снижению точности математических нейросетевых моделей.

В работах [56, 63] рекомендуется другой способ определения значимости входных параметров нейросетевых моделей. Его суть заключается в том, что из нейронной сети поочередно исключаются входные нейроны и осуществляется поочередное обучение таких усеченных сетей. Затем строится гистограмма, показывающая распределение погрешностей обобщения нейросетей по каждому исключенному входному параметру (входному нейрону). Понятно, что чем выше оказывается эта погрешность, тем значимее соответствующий исключенный входной параметр.

В заключение отметим, что вопрос о преимуществах той или иной технологии часто является предметом длинных и ни к чему не приводя-

щих дискуссий. Специалисты, глубоко освоившие одну из технологий, выступают с критикой другой технологии, и наоборот.

С точки зрения автора книги, вряд ли стоит противопоставлять эти две технологии. Опыт показывает, что для успешного достижения целей полезно пользоваться всеми доступными технологиями, независимо от их идеологической базы.

9.4. НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ И МЕТОД МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

Ранее мы уже говорили о том, что метод математического моделирования является одним из самых мощных и перспективных современных инструментов получения научных и практических знаний, что он сыграл огромную роль в формировании нашей цивилизации и что эта роль в последнее время все более усиливается.

Следует заметить, что сам метод математического моделирования, как таковой, возник много веков назад одновременно с математикой. Собственно говоря, цель математики всегда состояла в том, чтобы отражать (т. е. моделировать) на своем специфическом языке закономерности окружающего мира.

Грандиозный толчок развитию метода математического моделирования во второй половине XX в. дало появление компьютерных технологий. Помимо того что компьютер взял на себя «черновую» работу, связанную с обработкой и визуализацией информации, он еще и научился выполнять такие интеллектуальные этапы математического моделирования, как, например, поиск решений математических уравнений.

Однако при всем этом за человеком оставался самый первый этап, связанный с построением, или, как говорят, с формулировкой математических моделей. За человеком оставалось выявление значащих (влияющих на результат моделирования) входных параметров модели. За человеком оставалось решающее слово о том, какие законы природы, какие закономерности изучаемой предметной области (объекта, процесса, явления) следует заложить в математическую модель.

Кстати, знание законов природы — это одно из основных достояний нашей цивилизации. Законы природы открывались учеными в творческих поисках, экспериментах и научных дискуссиях на протяжении всей истории человечества.

И вот, в конце XX—начале XXI в. мы становимся свидетелями события, значение и масштабы последствий которого в истории цивилизации мы пока до конца еще не осознали. Мы становимся свидетелями того, как эти «святые», высокоинтеллектуальные, сугубо «человеческие» функции берет на себя компьютер.

Оказывается, он способен заменить человека и здесь.

Оказывается, он может (страшно подумать!) совершать научные открытия.

Он может открывать законы природы и сам закладывать их в математические модели!

Другими словами, компьютер теперь может не только помогать человеку реализовывать его идеи в виде математических моделей, но и сам создавать математические модели. Он теперь может полностью осуществлять весь процесс математического моделирования от «А» до «Я». Причем во многих случаях компьютер справляется с созданием математических моделей лучше, чем человек. Например, компьютер может выявлять и учитывать в создаваемых им моделях закономерности и внутренние связи, не известные специалистам (см. гл. 5). В результате модели, создаваемые компьютером, учитывают большее количество факторов и закономерностей, влияющих на результат моделирования, что, конечно же, улучшает качество моделей.

Что же это за компьютер, который может не только обрабатывать математические формулы и решать математические уравнения, но и самостоятельно формулировать эти формулы и уравнения, извлекать их из анализа предметных областей?

Как мы убедились, изучив наш курс, это не совсем обычный компьютер. Все, что может обычный (фон-неймановский) компьютер — это складывать и вычитать двоичные числа. Как известно, более сложные операции, такие как умножение, деление и возведение в степень, обычный компьютер сводит к сложению и вычитанию, как и все остальное, что он вообще умеет делать. Согласитесь, что это совсем не то, что происходит в мозге ученого, когда он совершает научные открытия.

Открывать законы *природы* способен только *человек*, точнее — его *мозг*.

Значит, *компьютер* должен быть адекватен *мозгу*.

Значит, *компьютер* должен быть создан, как сказано в Библии, «*по образу и подобию*», но только не *Бога*, а *человеческого мозга*.

Здесь уместно процитировать слова из Библии: «И сказал Бог: сотворим человека по образу Нашему и по подобию Нашему...»[1)].

Именно идея создания компьютеров и компьютерных программ, работающих подобно человеческому мозгу, и лежит в основе одного из самых увлекательных и плодотворных разделов искусственного интеллекта — нейроинформатики.

Нейрокомпьютер (или его программная реализация — нейронная сеть), как и человеческий мозг, состоит из множества соединенных между собой элементарных ячеек — искусственных нейронов. Принцип их действия во многом аналогичен принципу действия биологических нейронов, из которых состоит мозг. Как и в мозге, нейроны нейрокомпьютера соединены между собой множеством связей, через которые происходит обмен электрическими сигналами. Как и биологические, нейроны нейрокомпьютера могут переходить в возбужденное состояние,

---

[1)] Библия. Книги Ветхого Завета, Бытие 1:26.

вырабатывая и посылая электрические сигналы другим нейронам. Причем переход в возбужденное состояние каждого нейрона тем вероятнее, чем большее количество сигналов он получил от других нейронов.

Согласно распространенным нейрофизиологическим представлениям, вся информация, хранящаяся в мозге, закодирована в виде матрицы сил синаптических связей. Она представляет собой множество значений электропроводности проводников, соединяющих биологические нейроны. Эти значения по мере жизнедеятельности мозга постоянно меняются, чем обеспечивается накапливание и корректировка хранящейся в мозге информации. Подобный механизм в нейрокомпьютере имитируется путем соответствующего изменения синаптических весовых коэффициентов, выполняющих роль электропроводности соединяющих нейроны проводников. Таким образом, в нейрокомпьютере, как и в мозге, вся информация представляется и хранится в виде матрицы сил синаптических связей нейронов.

В отличие от обычного компьютера нейрокомпьютер не программируется, а подобно человеку обучается; при этом происходит корректировка сил синаптических связей. Обучение проводится на обучающих примерах, содержащих информацию о моделируемой предметной области. Например, чтобы нейрокомпьютер «открыл» теорему Пифагора, надо нарисовать несколько различных прямоугольных треугольников, измерить длины их гипотенуз и катетов и передать эту информацию нейрокомпьютеру. Если измерения выполнены правильно и треугольников было достаточно много, то можно не сомневаться, что нейрокомпьютер повторит подвиг великого геометра древности!

Поступая аналогичным образом и собирая информацию о соответствующих предметных областях, можно при помощи нейрокомпьютера заново открыть закон Архимеда, законы Ньютона, все фундаментальные законы физики, химии, биологии, экономики и всех других наук.

Таким образом, нейрокомпьютерные и нейросетевые технологии, по существу, открывают принципиально новый подход к самой методике построения компьютерных математических моделей.

Появилась возможность строить математические модели, которые, исходя из одного только эмпирического опыта — обучающих примеров, сами извлекают закономерности предметных областей и позволяют эффективно использовать эти модели для решения широкого круга практических задач. При этом человеку — создателю нейрокомпьютера — не обязательно задумываться над законами физики, химии, биологии, экономики и т. д., нейрокомпьютер обнаруживает их сам.

С появлением нейросетевых и нейрокомпьютерных технологий появился новый способ построения математических моделей. Появился новый способ получения научных знаний, новый инструмент интеллектуального анализа данных, позволяющий выявлять ранее не известные, еще не исследованные зависимости и закономерности и активно использовать их для решения практических задач.

9.5. ФИЛОСОФСКИЕ ПРОБЛЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Прежде чем обсуждать философские проблемы искусственного интеллекта, вспомним еще раз краткую историю его появления и развития.

Исторически первой была высокоуровневая стратегия искусственного интеллекта, в основе которой лежал явный способ представления знаний, формируемых человеком-экспертом. Эта стратегия безраздельно господствовала начиная с XIII в. до середины XX в. Появление в 1940-х гг. низкоуровневой стратегии — создание нейрокомпьютера, получающего знания в процессе обучения, и его первые успехи были встречены с надеждами и энтузиазмом, который, однако, продержался недолго. К середине 1980-х гг. подавляющее большинство систем искусственного интеллекта представляли собой экспертные системы, ориентированные на конкретные предметные области. Однако победа высокоуровневой стратегии была временной. Конец XX в.—начало XXI в. ознаменовались широкомасштабным наступлением нейросетевых и нейрокомпьютерных технологий. Эти технологии пришли не только в новые предметные области, но и стали вытеснять экспертные системы из областей, где они традиционно применялись ранее. И вот уже в течение полувека между высоко и низкоуровневой стратегиями идет конкуренция.

Третья стратегия искусственного интеллекта, появившаяся на свет в середине 1970-х гг., навеяна Дарвиновской теорией возникновения и эволюции жизни на Земле. Эта стратегия предполагает поиск источника интеллекта в самом процессе эволюции и взаимодействия особей. Знания в эволюционных моделях приобретаются в ходе конкурентной борьбы между особями.

На сегодняшний день безусловным лидером как в области теоретических исследований, так и в части практических приложений являются нейросетевые и нейрокомпьютерные технологии. Попытки моделирования мозга на самом низком уровне абстрагирования — на уровне нейронной структуры — показывают себя как наиболее плодотворные технологии создания интеллектуальных систем.

Искусственный интеллект сегодня является лидирующей научной отраслью, поэтому как никакая другая дисциплина он нуждается в философском осмыслении результатов, парадигм, стратегий, методов. Как и всякая научная область, искусственный интеллект имеет свои корни в философии. Корни высокоуровневой стратегии искусственного интеллекта можно найти, например, в аристотелевской «Логике», а в полном объеме формальная аксиоматизация логических рассуждений представлена в работах таких великих философов, как Готлоб Фреге, Бертран Рассел, Курт Гедель и др.

Однако работы великих философов прошлого, разумеется, не охватывают коренных изменений, которые искусственный интеллект претерпел совсем недавно, т. е. в конце XX — начале XXI вв. К таким коренным

изменениям в первую очередь относится создание нейрокомпьютера, построенного по образу и подобию человеческого мозга. Именно появление нейрокомпьютера и феномен триумфального практического внедрения нейросетевых и нейрокомпьютерных технологий менее всего исследованы философами, но более всего в этом нуждаются.

Для философской оценки последствий появления нейрокомпьютера и связанных с этим событий попробуем обратиться к широко известной библейской легенде, согласно которой Бог создал Адама и Еву по своему «образу и подобию». Создал, по-видимому, с определенной целью, ожидая от них определенных качеств и результата. Однако в последующем люди обнаружили несколько иные качества, чего Создатель, вероятно, не ожидал. И, согласно библейским преданиям, за одно из таких незапланированных и непредвиденных качеств Адам и Ева были выдворены из Рая…

Обратимся теперь к истории создания нейрокомпьютера. Как отмечает создатель первого нейрокомпьютера Ф. Розенблатт [48, 102], персептрон (нейрокомпьютер) был построен «по образу и подобию» человеческого мозга и предназначался для решения задачи узнавания букв латинского алфавита, причем только тех, которым он обучался. И эта цель Розенблаттом была успешно достигнута. Однако, наблюдая за последующими событиями, можно заключить, что со временем нейрокомпьютер стал проявлять такие свойства, которых его создатель (Ф. Розенблатт) от него не ожидал. Вот некоторые из них.

1. Способность к *обобщению* (см. § 4.4) — персептрон, обученный распознавать определенное количество образов, вдруг оказывается способным распознавать не только те образы, которым его обучали, но и такие, которых в обучающем множестве не было, т. е. образы, которых он никогда «не видел».

2. *Интуиция,* или «*шестое чувство*» (см. §§ 5.3, 5.9), — способность нейронных сетей и нейрокомпьютеров делать правильные выводы, руководствуясь информацией, которой для логического объяснения таких выводов недостаточно.

3. *Живучесть*. Известны случаи, когда люди продолжали совершать разумные поступки, потеряв в результате физической травмы до 30% объема мозга. То же самое происходит с нейрокомпьютерами: при выводе из строя отдельных нейрочипов нейрокомпьютеры продолжают решать поставленные задачи.

4. Свойства невербальности, гиперразмерности (§ 5.9), а также способность к извлечению знаний из данных также можно отнести к свойствам, заимствованным нейронными сетями и нейрокомпьютерами от мозга.

Отметим еще раз, что перечисленные свойства нейронных сетей и нейрокомпьютеров не предусматривались их создателями заранее, а обнаружились в процессе эксплуатации. Эти свойства появились сами по себе вследствие того, что нейросети и нейрокомпьютеры

являются моделями мозга. Они выполнены «по образу и подобию» человеческого мозга, поэтому нет ничего удивительного в том, что они (как и в случае примера из Библии) на вполне законных основаниях наследуют свойства своего прототипа.

Вышеприведенные рассуждения и факты свидетельствуют о том, что между ***нейрокомпьютером*** как моделью ***человеческого мозга*** и ***человеком*** как «моделью ***Бога***» существует некая аналогия. Возможно, эту аналогию можно взять за основу при прогнозировании последствий развития искусственного интеллекта и интеллектуальных систем. Возможно, искусственная модель человека будет и дальше обнаруживать непредусмотренные создателями свойства, и между моделями и их прототипами — людьми — могут возникнуть непонимание и проблемы, подобные тем, которые описаны в Библии…

9.6. ПРОГНОЗЫ НА БУДУЩЕЕ

Трудно назвать какую-либо другую научную область, которая могла бы сравниться с искусственным интеллектом по изобилию прогнозов ее будущих успехов. Ученые строили прогнозы возможностей искусственного интеллекта во все времена, связывая эти прогнозы с будущим человечества. Причем далеко не всегда человеческое общество воспринимало прогнозы ученых с благодарностью. Взять хотя бы основателя искусственного интеллекта Раймунда Луллия (рис. 9.1), поплатившегося жизнью за свои смелые убеждения (см. § 1.1)

Особое место в серии прогнозов будущего искусственного интеллекта занимают знаменитые предсказания нобелевского лауреата Герберта Саймона (рис. 9.2), опубликованные в 1957 г.:

— ***В ближайшее десятилетие ЭВМ завоюет титул чемпиона мира по шахматам.***

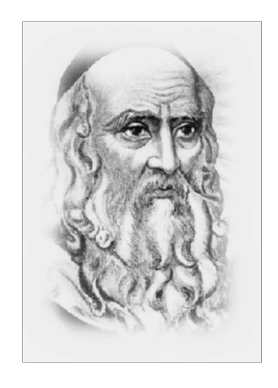


Рис. 9.1. Основатель искусственного интеллекта Раймунд Луллий (XIII–XIV вв.)



Рис. 9.2. Нобелевский лауреат Герберт Саймон (XX в.)

— ***В пределах десяти лет ЭВМ откроет и сумеет доказать новую важную математическую теорему.***
— ***В десятилетний срок большинство теорий в области психологии примет вид программ для вычислительной машины.***

С тех пор прошло более полувека, и мы можем констатировать, что два первых предсказания, хотя и с большим опозданием, но сбылись. Компьютерная программа «Deep Blue» однажды выиграла матч у чемпиона мира по шахматам, а программа «Логик-теоретик» сформулировала две новые теоремы алгебры логики. Однако о третьем предсказании Саймона мы ничего подобного пока сказать не можем, если не считать некоторые весьма скромные попытки применения методов искусственного интеллекта в психологических проблемах (например [61, 67, 76, 78, 79]), а также психодиагностические программы сайта **www.PermAI.ru**.

Ажиотаж вокруг искусственного интеллекта и серия радужных прогнозов журналистов и писателей-фантастов в середине XX в., вызванные первыми успехами персептрона, закончились провалом американского военно-стратегического проекта «Свой—Чужой» и объявлением искусственного интеллекта тупиковым научным направлением (см. § 4.6). Не прибавили авторитета искусственному интеллекту и нескончаемые споры философов на тему «Может ли машина мыслить?», популярные в конце XX в. и продлившиеся вплоть до настоящего времени.

Что происходит сейчас?

Нейроинформатика вышла из научного и прикладного тупика 1960–1970-х гг. Сейчас искусственный интеллект является одной из самых востребованных и быстро развивающихся научных областей.

Искусственный интеллект стал модным, и это настораживает.

Настораживают громкие прогнозы, которые делают современные отечественные ученые (рис. 9.3), объединившиеся в общественное движение «Россия 2045».

Цель этого движения: «*К 2045 году сделать человека бессмертным!*».






Рис. 9.3. Лидеры движения «Россия-2045» (слева-направо): Д. И. Дубровский, В. И. Аршинов, В. Л. Дудин-Барковский, А. Я. Каплан (XXI в.)



Рис. 9.4. Технический директор программ искусственного интеллекта компании Google Р. Курцвейл (XXI в.)

Не менее смелые предсказания сделал Рэй Курцвейл (рис. 9.4) — технический директор программ искусственного интеллекта в компании Google. В 2005 г. он опубликовал серию прогнозов на 2099 г. Вот некоторые из них.

— *Естественный процесс мышления не будет иметь преимущества по сравнению с машинным. Системы искусственного интеллекта получат равные права с человеком. Разница между человеком и машиной исчезнет.*
— *Системы искусственного интеллекта превзойдут по численности естественных людей.*
— *Несмотря на недостатки естественных людей, системы искусственного интеллекта уважают их как своих прародителей.*

Мы воздержимся от обсуждения приведенных выше прогнозов, а вместо этого изложим собственное видение будущих успехов систем искусственного интеллекта, основываясь на наших собственных результатах, которыми мы реально располагаем уже сейчас.

По нашему мнению, в ближайшие десять лет реально наступление следующих событий.

1. *Реально создание интеллектуальных медико-диагностических систем, превосходящих по точности постановки диагнозов естественных врачей. Реально выявление с помощью интеллектуальных систем новых неизвестных ранее медицинских знаний и использование этих знаний для улучшения качества медицинской практики.*

Одна из таких интеллектуальных систем — «Система диагностики и прогнозирования заболеваний сердечно-сосудистой системы» — выложена в свободном доступе на сайте **www.PermAI.ru**. С помощью нее выявлены новые неизвестные ранее знания (см. § 5.2, а также [69, 108, 109]), которые используются для совершенствования существующей практики лечения и профилактики заболеваний сердца.

2. *Реально создание интеллектуальных систем диагностики сложных технических устройств, превосходящих по своим возможностям и точности известные инженерные методики. Их внедрение в инженерную практику позволит повысить надежность технических устройств и, таким образом, способствовать решению проблемы снижения техногенных аварий и катастроф — актуальной проблемы XXI в.*

Наш опыт разработки нейросетевых систем диагностики авиационных двигателей показал (см. § 5.3, а также [30]), что интеллектуальные системы позволяют выявлять такие дефекты, которые обычными инженерными методами выявлены быть не могут.

3. *Реально создание интеллектуальных систем, способных получать точные аналитические решения краевых задач математической физики, что позволит преодолеть современный кризис прикладной математики* (см. § 8.4., а также [12, 68, 77, 80, 86]). *Реально внедрение таких интеллектуальных систем в инженерную практику, что также будет способствовать снижению числа техногенных аварий и катастроф.*

С нашего сайта **www.PermAI.ru** можно обратиться к демонстрационному прототипу интеллектуальной системы «Искусственный математик», способной получать точные аналитические решения краевых задач теории упругости, теплопроводности, термоупругости и др.

4. *Реально создание интеллектуальных систем, прогнозирующих экономическое состояние предприятий и позволяющих разрабатывать рекомендации по оптимизации их деятельности.*

В работе [70] описан опыт создания нейросетевой системы прогнозирования вероятности банкротства российских банков. Помимо прогнозов, система позволяет получать полезные рекомендации по предотвращению банкротства конкретных банков. Программа выложена на сайте **www.PermAI.ru**.

5. *Реально создание интеллектуальных систем, предназначенных для прогнозирования развития политических событий и влияния на эти события.*

В § 5.6 описан случай успешного прогнозирования победы Д. А. Медведева на президентских выборах 2008 г., выполненного за полтора года до этих выборов [71]. Там же (см. также [82]) приведены подтвердившиеся впоследствии результаты прогнозирования рейтингов известных политических деятелей, а также рекомендации по улучшению этих рейтингов.

6. *Реально создание и применение интеллектуальных систем в области криминалистики.*

В работе [73] описан опыт создания нейросетевого детектора лжи, значительно превосходящего известные штатные аналоги по точности заключений.

В работе [61] изложен опыт создания интеллектуальной системы, предназначенной для изучения личности и выявления серийных убийц. Программа, оценивающая склонность человека к насилию, выложена на сайте **www.PermAI.ru**.

7. *Реально создание интеллектуальных систем, предназначенных для прогнозирования результатов спортивных состязаний и для оптимизации программы подготовки спортсменов с целью получения ими наивысших спортивных результатов.*

В нашей практике нейронные сети уже применялись для прогнозирования результатов Олимпийских игр 2014 г. [65, 83], чемпионата мира 2015 г. [74, 75], а также при разработке рекомендаций для известных спортсменов: фигуриста Е. В. Плющенко [65] и спортсменов-бегунов [74].

8. *Реально создание и широкое применение интеллектуальных систем в психологии*.

Напомним третье предсказание нобелевского лауреата Герберта Саймона:

— В десятилетний срок большинство теорий в области психологии примет вид программ для вычислительной машины.

Мы знаем, что два первых предсказания, хотя и с опозданием, но сбылись. Однако о третьем предсказании Саймона мы ничего подобного пока сказать не можем. Почему?

Пытаясь решать задачи психодиагностики, психологи выдвигают различные гипотезы, называют в качестве наиболее важных те или иные факторы, влияющие, по их мнению, на способности и предрасположенности человека. Они вводят такие труднодоступные для количественной оценки понятия, как «сила воли», «сила характера» и др., разрабатывают различные тесты для их измерения. Естественно, что при сборе статистической информации возникают проблемы, особенно если речь идет о человеке, жившем в прошлых веках. В то же время, психологи обычно не обращают внимания на легкодоступную информацию о людях, известную, например, из астрологических или нумерологических баз, отвергая ее как лженаучную.

По-видимому, психологи еще не полностью освоили и оценили возможности современных методов интеллектуального анализа данных, которые позволяют на начальных стадиях создания интеллектуальных систем в качестве входных параметров вводить любое количество любых факторов. Затем в ходе исследований математические модели позволяют выяснить, какие из этих параметров действительно влияют на результат моделирования, а какие — нет, вне зависимости от того, признаются эти факторы официальной наукой или нет.

Заметим, что наш небольшой опыт применения нейросетевых технологий для создания интеллектуальных систем, выявляющих способности и предрасположенности человека, оказался весьма успешным. Так, в работе [78] описана интеллектуальная система, выявляющая способность человека к руководящей деятельности, в [3] — к научной деятельности, в [84] — к бизнесу. В [76] описана интеллектуальная система, способная прогнозировать степень успешности будущей карьеры студентов вуза, что позволяет оптимальным способом подбирать траектории их обучения. В [67] приведен опыт создания системы, определяющей предрасположенность подростков к наркотической зависимости, а также позволяющей получать индивидуальные рекомендации по снижению такой зависи-

мости. В [61] — предрасположенность человека к насилию. Имеются программы, оценивающие предрасположенность человека к суициду, анорексии и др. Все программы находятся в свободном доступе в разделе «Проекты» сайта **www.PermAI.ru**. Помимо количественной оценки способностей, склонностей и предрасположенностей людей, с помощью этих программ можно получать полезные рекомендации по их развитию или, наоборот, — нивелированию.

Таким образом, у нас есть все основания надеяться, что третье предсказание Герберта Саймона о широком применении компьютерных программ в психологии в скором времени также сбудется.

9. ***Реально применение методов искусственного интеллекта в областях, выходящих за рамки традиционных наук. Реально получение новых, ранее неизвестных, но полезных для практики знаний, в том числе таких, которые меняют наше представление о мире. Можно надеяться, что методы искусственного интеллекта помогут наконец разгадать загадки эзотерики, уфологии, астрологии и других учений, выходящих за рамки официального миропонимания.***

Как уже отмечалось, на нашем сайте **www.PermAI.ru** выложены интеллектуальные системы, позволяющие ставить диагнозы медицинских заболеваний, выявлять способности людей к бизнесу, к научной деятельности, к руководящей деятельности, склонности к наркомании, суициду, анорексии, насилию и другим порокам. Во многих случаях низкая погрешность результатов, зафиксированная при тестовых испытаниях программ, обеспечивается тем, что, помимо традиционных входных параметров, учитываются еще и такие параметры, как солнечная активность, фазы луны, градусы небесных тел в момент рождения человека и др. Природу влияния таких параметров пока не удается объяснить в рамках современного уровня развития науки. Но это не мешает их использовать при создании эффективных и весьма полезных для практики интеллектуальных систем.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Постарайтесь дать определение человеческого интеллекта и искусственного интеллекта.
2. В чем разница между сильным и слабым искусственным интеллектом?
3. Перечислите уровни абстракции, на которых разрабатываются теории искусственного интеллекта.
4. Перечислите преимущества и недостатки нейросетевых технологий по сравнению с технологией экспертных систем.

5. Что общего между технологией регрессионного анализа и нейросетевыми технологиями? Какие технологии более универсальны?
6. Какие из стратегий искусственного интеллекта являются лидирующими?
7. Какие свойства унаследовали нейронные сети от своего прототипа — человеческого мозга?
8. Почему, по вашему мнению, не сбылось третье предсказание Г. Саймона?
9. Каким вам представляется будущее искусственного интеллекта?

СПИСОК ЛИТЕРАТУРЫ

1. *Александров В. В. Стенчиков Г. Л.* Об одном вычислительном эксперименте, моделирующем последствия ядерной войны // Вычислительная математика и математическая физика. 1984. Т. 24. С. 140–144.
2. *Арнольд В. И.* О функциях трех переменных // Доклады Академии Наук СССР. 1957. Т. 114. № 4.
3. *Байдин Д. Ю., Макурина Т. В., Черепанов Ф. М., Ясницкий Л. Н.* Нейросетевая система прогнозирования склонности к научной деятельности // Актуальные проблемы математики, механики, информатики: сб. тез. Научню-практ. Конф. (Пермь, 30 октября — 1 ноября 2012 г.) / Перм. Гос. Нац. Исслед. Ун-т. — Пермь, 2012. С. 153.
4. *Ванюшин В. А., Галушкин А. И., Тюхов Б. П.* Построение и исследование оптимальных многосложных систем распознавания образов в режиме обучения // Сб. «Некоторые вопросы биологической кибернетики» под ред. акад. Берга А. Л. — М.: Наука: 1972. С. 315–323.
5. *Гаврилова Т. А., Хорошевский В. Ф.* Базы знаний интеллектуальных систем. — СПб.: Питер, 2001.
6. *Галушкин А. И., Зак А. С., Тюхов Б. П.* К сравнению критериев оптимизации адаптивных систем распознавания образов // Кибернетика. 1970. № 5. С. 122–130.
7. *Галушкин А. И.* Нейронные сети: основы теории. — М: Горячая линия—Телеком, 2012. 496 с.
8. *Галушкин А. И.* Об алгоритмах адаптации в многослойных системах распознавания образов // Доклады АН УССР (представлено акад. Глушковым В. М.). 1973. Т. 91. № I, С. 15–21.
9. *Галушкин А. И., Симоров С. Н.* Нейросетевые технологии в России (1982–2010). — М.: Горячая линия-Телеком, 2011. 316 с.
10. *Галушкин А. И.* Синтез многослойных систем распознавания образов. — М.: «Энергия», 1974. 367 с.
11. *Галушкин А. И.* Теория нейронных сетей. — М.: Издательское предприятие редакции журнала «Радиотехника», 2000.
12. *Гладкий С. Л., Степанов Н. А., Ясницкий Л. Н.* Интеллектуальное моделирование физических проблем. — Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2006. 200 с.
13. *Горбань А. Н.* Нейроинформатика и ее приложения // Открытые системы. 1998. № 04–06.

14. *Горбань А. Н., Миркес Е. М.* Логически прозрачные нейронные сети. Нейроинформатика и ее приложения: Тез. Докл. III Всесоюз. семинара. — Красноярск: Изд-во КГТУ, 1999.

15. *Горбань А. Н., Миркес Е. М.* Нейронные сети ассоциативной памяти, функционирующие в дискретном времени / Вычислительный центр СО РАН. Красноярск. Рукопись деп. В ВИНИТИ 17.07.97. № 2436-В97.

16. *Горбань А. Н.* Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сиб. журнал выч. матем. 1998. Т. 1. № 1. С. 12–24.

17. *Горбань А. Н.* Обобщенная аппроксимационная теорема и точное представление многочленов от нескольких переменных суперпозициями многочленов от одного переменного // Изв. вузов. Математика. 1998. № 5. С. 6–9.

18. *Горбань А. Н.* Обучение нейронных сетей. — М.: СП ПараГраф, 1990. 159 с.

19. *Дайтбегов Д. М.* Компьютерные технологии анализа данных в эконометрике, 2008.

20. *Елтышев В. А.* Напряженно-деформированное состояние оболочечных конструкций с наполнителем. — М.: «Наука», Москва, 1981.

21. *Заенцев И. В.* Нейронные сети: основные модели. — Воронеж: Изд-во Воронеж. ун-та, 1999.

22. *Иванов А. И.* Подсознание искусственного интеллекта: программирование автоматов нейросетевой биометрии языком их обучения. — Пенза: Электронная книга издательства ОАО «ПНИЭИ». 2012. 125 с.

23. Информатика: Учебник / Под ред. Н. В. Макаровой. — М.: Финансы и статистика, 1997.

24. Искусственный интеллект. Кн. 1. Системы общения и экспертные системы / Под ред. Э. В. Попова. — М.: Радио и связь, 1990.

25. Искусственный интеллект. Кн. 2. Модели и методы / Под ред. Д. А. Поспелова. — М.: Радио и связь, 1990.

26. *Каллан Р.* Основные концепции нейронных сетей / Пер. с англ. — М: Вильямс, 2001. 288с.

27. *Колмогоров А. Н.* О представлении непрерывных функций нескольких переменных суперпозицией непрерывных функций меньшего числа переменных // Доклады АН СССР. — М.: Изд-во Академии наук СССР, 1956. Т. 108.

28. *Колмогоров А. Н.* О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения // Доклады АН СССР. — М.: Изд-во АН СССР, 1957. Т. 114.

29. *Комарцова Л. Г., Максимов А. В.* Нейрокомпьютеры. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2002.

30. *Конев С. В., Сичинава З. И. Ясницкий Л. Н.* Применение нейросетевых технологий для диагностики неисправностей авиационных двигателей. Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2005. № 2. С. 43–47.

31. *Круглов В. В., Борисов В. В.* Искусственные нейронные сети. Теория и практика. — М.: Горячая линия — Телеком, 2002.

32. *Лаврентьев М. А., Шабат Б. В.* Проблемы гидродинамики и их математические модели. — М.: Наука, 1973. 416 с.

33. *Люгер Дж. Ф.* Искусственный интеллект: Стратегии и методы решения проблем / Пер. с англ. — М.: Вильямс, 2003.

34. *Минский М., Пейперт С.* Персептроны. — М.: Мир, 1971.

35. *Мкртчян С. О.* Нейроны и нейронные сети. Введение в теорию формальных нейронов. — М.: Энергия, 1971.

36. *Мурашов Д. И., Ясницкий Л. Н.* Социальный генетический алгоритм // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2006. № 4. С. 53–60.

37. Нейроинформатика / А. Н. Горбань, В. Л. Дунин-Барковский, А. Н. Кирдин и др. — Новосибирск: Наука, 1998.

38. Нейрокомпьютерная парадигма и общество / Под ред. Ю. Ю. Петрунина. — М.: Издательство Московского университета. 2012.

39. Нейрокомпьютеры и их применение. Кн. 5. Нейронные сети: история развития теории / Под ред. А. И. Галушкина и Я. З. Цыпкина. — М.: Радиотехника, 2001.

40. Обработка знаний / Под ред. С. Осуги. — М.: Мир, 1989. 293 с.

41. *Осовский С.* Нейронные сети для обработки информации / Пер. с польск. — М.: Финансы и статистика, 2002.

42. *Пенский О. Г., Зонова П. О., Муравьев А. Н. и др.* Гипотезы и алгоритмы математической теории исчисления эмоций: монография. — Пермь: Пермский государственный университет, 2009. 152 с.

43. *Пенский О. Г.* Математические модели эмоциональных роботов: монография. — Пермь: Пермский государственный университет, 2010. 192 с.

44. *Попов Э. В.* Общение с ЭВМ на естественном языке. — М.: Наука, 1982.

45. Представление знаний / Под ред. С. Осуги, Ю. Саэки. — М.: Мир, 1990.

46. Представление и использование знаний / Под ред. Х. Уэно, М. Исидзуки. — М.: Мир, 1987.

47. *Рассел С.* Искусственный интеллект: современный подход / С. Рассел, П. Норвинг ; пер. с англ. — М: Издат. дом «Вильямс», 2006. 1408 с.

48. *Розенблатт Ф.* Принципы нейродинамики. Перцептроны и теория механизмов мозга. — М.: Мир, 1965. 480 с.

49. *Савельев А. В.* Нейронанофизическая природа распределенного молекулярно-клеточного наноакцептора результата действия — аналитическое нейрокомпьютерное моделирование // Нейрокомпьютеры: разработка, применение. 2013. № 6. С. 35–41.

50. *Терехов С. А., Воленко Е. В., Квичанский А. В., Щукин Н. В.* Генетическая оптимизация нейронных сетей. — Снежинск: Препринт ВНИИТФ 114, 1997.

51. *Уоссермен Ф.* Нейрокомпьютерная техника. — М.: Мир, 1992.

52. *Уотермен Д.* Руководство по экспертным системам. — М.: Мир, 1989.

53. *Фогель Л., Оуэнс М.* Искусственный интеллект и эволюционное моделирование. — М.: Мир, 1969.

54. *Хайкин С.* Нейронные сети: Полный курс. — 2-е изд. / Пер. с англ. — М.: Вильямс, 2006. 1104 с.

55. *Хеннер Е. К.* Информатика. — М.: Академия, 2003. 816 с.

56. *Черепанов Ф. М.* Симулятор нейронных сетей для вузов // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2012. № 3. С. 98–105.

57. *Черепанов Ф. М., Ясницкий Л. Н.* Нейросетевой фильтр для исключения выбросов в статистической информации // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2008. № 4. С. 151–155.

58. *Черепанов Ф. М., Ясницкий Л. Н.* Нейросимулятор 5.0. Свидетельство о государственной регистрации программы для ЭВМ № 2014618208. Заявка Роспатент № 2014614649. Зарегистрировано в Реестре программ для ЭВМ 12 августа 2014 г.

59. *Эндрю А.* Искусственный интеллект. — М.: Мир, 1985. 265 с.

60. *Ясницкий Л. Н., Богданов К. В., Черепанов Ф. М.* Технология нейросетевого моделирования и обзор работ Пермской научной школы искусственного интеллекта // Фундаментальные исследования. 2013. № 1 (ч. 3). С. 736–740.

61. *Ясницкий Л. Н., Ваулева С. В., Сафонова Д. Н., Черепанов Ф. М.* Использование методов искусственного интеллекта в изучении личности серийных убийц // Криминологический журнал Байкальского государственного университета экономики и права. 2015. Т. 9, № 3. С. 423–430.

62. *Ясницкий Л. Н.* Введение в искусственный интеллект. — Пермь: Изд-во Пермского университета, 2001. 143 с.

63. *Ясницкий Л. Н.* Введение в искусственный интеллект. — М.: Издательский центр «Академия», 2005. 176 с.

64. *Ясницкий Л. Н.* Введение в искусственный интеллект. — 3-е изд. — М.: Издательский центр «Академия», 2010. 176 с.

65. *Ясницкий Л. Н., Внукова О. В., Черепанов Ф. М.* Прогноз результатов Олимпиады-2014 в мужском одиночном фигурном катании методами искусственного интеллекта // Современные проблемы науки и образования. — 2014. № 1; URL: **http://www.science-education.ru/115-11339** (дата обращения: 25.12.2013).

66. *Ясницкий Л. Н.* Гильберт, Колмогоров, Арнольд, искусственный интеллект и современный кризис прикладной математики (К 70-летию со дня рождения В. И. Арнольда) // Вопросы искусственного интеллекта. 2008. № 1. С. 77–80.

67. *Ясницкий Л. Н., Грацилев В. И., Куляшова Ю. С., Черепанов Ф. М.* Возможности моделирования предрасположенности к наркозависимости методами искусственного интеллекта. Вестник Пермского университета. Серия: Философия. Психология. Социология. 2015. № 1(21). С. 61–71.

68. *Ясницкий Л. Н., Данилевич Т. В.* Современные проблемы науки. — М.: БИНОМ. Лаборатория знаний, 2011. 294 с.

69. *Ясницкий Л. Н., Думлер А. А., Богданов К. В., Полещук А. Н., Черепанов Ф. М., Макурина Т. В., Чугайнов С. В.* Диагностика и прогнозирование течения заболеваний сердечно-сосудистой системы на основе нейронных сетей // Медицинская техника. 2013. № 3. С. 42–44.

70. *Ясницкий Л. Н., Иванов Д. В., Липатова Е. В.* Нейросетевая система оценки вероятности банкротства банков // Бизнес-информатика. 2014. № 3. С. 49–56.

71. *Ясницкий Л. Н.* Интеллектуальные информационные технологии и системы. — Пермь: Пермский государственный университет, 2007. 271 с.

72. *Ясницкий Л. Н.* Искусственный интеллект и новые возможности компьютерного моделирования // Вестник Пермского университета. Серия: Информационные системы и технологии. 2005. № 4. С. 81–86.

73. *Ясницкий Л. Н.* Искусственный интеллект. Элективный курс: Учебное пособие. — М.: БИНОМ. Лаборатория знаний, 2011. 240 с.

74. *Ясницкий Л. Н., Киросова А. В., Ратегова А. В., Черепанов Ф. М.* Методика прогнозирования результатов спортивных состязаний на примере чемпионата мира-2015 по легкой атлетике // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2014. № 3. С. 90–97.

75. *Ясницкий Л. Н., Кировоса А. В., Ратегова А. В., Черепанов Ф. М.* Прогноз результатов чемпионата мира-2015 по легкой атлетике методами искусственного интеллекта // Современные проблемы науки и образования. 2014. № 4; URL: **www.science-education.ru/118-14423** (дата обращения: 28.08.2014).

76. *Ясницкий Л. Н., Кузнецов А. Г., Селезнева С. М., Солохина А. Д., Тюлькина Д. В., Черепанов Ф. М.* Применение нейросетевых технологий в изучении акмеологического потенциала студентов вуза // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2014. № 4. С. 120–126.

77. *Ясницкий Л. Н.* Метод фиктивных канонических областей в механике сплошных сред. — М.: Наука, ФИЗМАТЛИТ, 1992. 128 с.

78. *Ясницкий Л. Н., Михалева Ю. А., Черепанов Ф. М.* Возможности методов искусственного интеллекта для выявления и использования новых знаний на примере задачи управления персоналом // International Journal of Unconventional Science. Журнал Формирующихся Направлений Науки. 2014. Вып. 6; URL: **http://www.unconv-science.org/n6/yasnitsky/** (дата обращения: 20.11.2014).

79. *Ясницкий Л. Н.* Нейронные сети — инструмент для получения новых знаний: успехи, проблемы, перспективы // Нейрокомпьютеры: разработка, применение. 2015. № 5. С. 48–56.

80. *Ясницкий Л. Н.* Обзор работ по развитию и применению метода фиктивных канонических областей в научных и инженерных проблемах // Современные проблемы науки и образования. 2012. № 4; URL: **http://www.science-education.ru/104-6786** (дата обращения: 27.07.2012).

81. *Ясницкий Л. Н.* Об одном способе решения задач теории гармонических функций и линейной теории упругости // Прочностные и гидравлические характеристики машин и конструкций. — Пермь. Изд. Пермского политехнического ин-та, 1973. С. 78–83.

82. *Ясницкий Л. Н.* О возможностях применения методов искусственного интеллекта в политологии // Вестник Пермского университета. Серия: Политология. 2008. № 2. С. 147–155.

83. *Ясницкий Л. Н., Павлов И. В., Черепанов Ф. М.* Прогнозирование результатов Олимпийских игр 2014 года в неофициальном командном зачете методами искусственного интеллекта // Современные проблемы науки и образования. 2013. № 6; URL: **http://www.science-education.ru/113-11206** (дата обращения: 25.12.2013).

84. *Ясницкий Л. Н., Порошина А. М., Тавафиев А. Ф.* Нейросетевые технологии как инструмент для прогнозирования успешности предпринимательской деятельности // Российское предпринимательство. 2010. № 4(2). С. 8–13.

85. *Ясницкий Л. Н.* Принципы построения экспертной системы для аналитического решения краевых задач // Математика программных систем. Межвузовский сборник научных трудов. — Пермь: Изд. Пермского государственного университета, 2001. С. 105–114.

86. *Ясницкий Л. Н.* Современный кризис прикладной математики и перспективы его преодоления // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2007. № 7. С. 192–197.

87. *Ясницкий Л. Н.* Удержаться «на плечах гигантов» (вводная статья) // Труды семинара «Компьютерные методы в механике сплошной среды». 2006–2007 гг. / Под ред. А. Л. Смирнова, Е. Ф. Жигалко. — СПб.: Изд-во СПб. Ун-та, 2008. — С. 3–15.

88. *Ясницкий Л. Н., Черепанов Ф. М.* Искусственный интеллект. Элективный курс: Методическое пособие по преподаванию. — М.: БИНОМ. Лаборатория знаний, 2011. 216 с.
89. *Ясницкий Л. Н., Черепанов Ф. М.* О возможностях применения нейросетевых технологий в политологии // Нейрокомпьютеры: разработка и применение. 2010. № 8. Вып. 4. С. 47–53.
90. *Ясницкий Л. Н., Черепанов Ф. М.* Применение нейросетевых технологий в политологии // Нейрокомпьютерная парадигма и общество / Под ред. Ю. Ю. Петрунина. — М.: Издательство Московского университета, 2012. С. 13–25.
91. *Fahlman S. E., Lebiere C.* The cascade-correlation learning architecture // Advances in NIP2 / Ed. D. Touretzky. — Morgan Kaufmann, 1990. P. 524–532.
92. *Golovko V., Savitsky Ju., Gladischuk V.* A neural net for prediction problems // Proceedings of Int. Conf. On technical informatics. Timisoara: University of Timisoara, Romania, 1996. P. 49–52.
93. *Haykin S.* Neural Networks: A Comprehensive Foundation // MacMillan College Publishing Co. New York, 1994.
94. *Hebb D. O.* The Organization of Behavior. John Wiley & Sons. New York, 1949.
95. *Hecht-Nielsen R.* Kolmogorov's Mapping Neural Network Existence Theorem // Proc. IEEE First Annual Int. Conf. On Neural Networks. San Diego, CA, 1987. Vol. 3. PP. 11–14.
96. *Hecht-Nielsen R.* Neurocomputing. — Amsterdam: Addison Wesley, 1991.
97. *Holland J. H.* Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.
98. *Hopfield J. J.* Neurons with graded response have collective computational properties like thous of two-state neurons. Proceedings of the National Academy of Sciences. 1984. No 81.
99. *Klimauskas G.* Neural Ware — User manual. Natick, USA: Neural Ware Inc., 1992.
100. *McCulloch W. S., Pitts W.* A Logical Calculus of Ideas Immanent in Nervous Activity // Bull. Mathematical Biophysics. 1943. V. 5.
101. *Rosenblatt F.* The perseptron: a probabilistic model for information storage and organization in the brain // Psychological Review. 1958. V. 65.
102. *Rosenblatt F.* Principles of Neurodynamics. Spartan Books. New York, 1962.
103. *Rummelhart D. E., Hilton G. E., Williams R. J.* Learning internal representations by error propagation. In McClelland et al. 1986.
104. *Seber G. A. F Wild C. J.* Nonlinear Regression. New York: John Wiley and Sons, 1989.
105. *Valtzev V. B., Grigoriev V. R., Nikonov V. G., Kobzar I. V.* Brainputer and Neural Architecture of Neurocomputers. Journal of Neurocomputer. 1996. Vol. 1. No. 1. P. 54–59.

106. *Werbos P.* Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences // Phd Thesis, Dept. of Applied Mathematics. Harvard University, Cambridge, Mass., 1974.

107. *Widrow B. Hoff M. E.* Adaptive switching circuits. 1960 IRE WESTCON Conferenction Record. New York, 1960.

108. *Yasnitsky L. N., Dumler A. A., Bogdanov K. V., Poleschuk A. N., Cherepanov F. M., Makurina T. V., Chugaynov S. V.,* Diagnosis and Prognosis of Cardiovascular Diseases on the Basis of Neural Networks // Biomedical Engineering. 2013. Vol. 47. No 3. PP. 160–163.

109. *Yasnitsky L. N., Dumler A. A., Poleshchuk A. N., Bogdanov C. V., Cherepanov F. M.* Artificial Neural Networks for Obtaining New Medical Knowledge: Diagnostics and Prediction of Cardiovascular Disease Progression // Biology and Medicine (Aligarh). 2015. 7(2): BM-095-15, 8 pages. (**http://www.biolmedonline.com/Articles/Vol7_2_2015/BM-095-15_Artificial-Neural-Networks-for-Obtaining-New-Medical-Knowledge-Diagnostics-and-Prediction-of-Cardiovascular-Disease-Progr.pdf**).

ПРИЛОЖЕНИЕ. ОТЗЫВЫ РЕЦЕНЗЕНТОВ

РЕЦЕНЗИЯ 1 НА КНИГУ Л. Н. ЯСНИЦКОГО «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ»

Автор рецензируемой книги хорошо известен специалистам своими учебными пособиями, монографиями и научными работами, посвященными развитию и применению интеллектуальных информационных технологий. Его учебное пособие «Введение в искусственный интеллект», имеющее три переиздания, пользуется популярностью среди студентов, ученых и преподавателей российских вузов, о чем свидетельствует большое количество цитирований этой книги. Его новая книга хотя и содержит часть старых авторских наработок, вошедших в предыдущие издания, однако написана с учетом быстро меняющихся реалий современной информатики.

Автор выдержал классическую структуру вузовского учебника.

Первая глава является вводной и посвящена историческим событиям, связанным с возникновением и становлением нового научного направления. Дается краткое (с соблюдением хронологии возникновения и развития) изложение современных технологий создания интеллектуальных систем и области их приложений.

Во второй главе дается популярное описание современных моделей представления знаний.

В третьей главе описана технология создания экспертных систем.

Четвертая, пятая, шестая и седьмая главы являются основными и наиболее полными. Показав в предыдущих главах преимущества и недостатки основных стратегий (технологий) создания интеллектуальных систем, автор концентрирует внимание на лидирующей в последнее время стратегии — технологии нейросетевого моделирования. Здесь подробно излагаются основополагающие идеи нейроинформатики, начиная от принципа действия математического нейрона, первых успехов распознавания букв и кончая современными проектами интеллектуального анализа данных. Подробно описываются как успехи, так и неудачи нейротехнологий, так что каждая новая идея, каждый новый алгоритм появляются не на пустом месте, а возникают закономерно — как единственный способ выхода из научного тупика. Тем самым учебник поддерживает принцип историзма, рассматривая явления

в динамике их изменения, становления во времени, в закономерном историческом развитии.

Несомненным преимуществом, придающим особый колорит книге, является то, что в качестве примеров применения нейросетевых технологий автор приводит свои собственные проекты, некогда придуманные, реализованные и внедренные под его руководством и при его участии.

Как справедливо отметил автор, нейроинформатика в настоящее время имеет недостаточную теоретическую базу. Успех применения нейросетевых технологий во многом зависит не только от знаний теоретического материала, но и от опыта и даже от интуиции исследователя. Сегодня в мире существует множество научных школ, которые развивают и применяют свои излюбленные нейросетевые парадигмы, свои собственные технологические приемы — так называемые ноу-хау, которые обычно остаются «за кадром». Преимущество книги состоит в том, что автор щедро делится собственными технологическими секретами, посвятив им целый подраздел. Практическим дополнением к книге служит специальный сайт, содержащий комплекс лабораторных работ и авторский нейропакет — удобный, легко осваиваемый программный инструментарий, предназначенный для создания, оптимизации и использования нейронных сетей.

Главы заканчиваются изложением новых неклассических нейросетевых парадигм, которые еще не нашли широкого применения и перспективность которых еще окончательно не выяснена.

Глава восьмая «Компьютерное творчество», на первый взгляд, выпадает из общей прагматической направленности учебника. Она начинается с изложения философских трактовок творчества, объясняет принципы моделирования музыкального и поэтического творчества, однако заканчивается серьезными проблемами моделирования научного творчества, в частности — возможностями моделирования деятельности математиков, выполняющих решение сложных краевых задач математической физики.

Девятая глава подводит итог учебного курса. Технологии создания интеллектуальных систем сравниваются между собой как в теоретическом, так и в прикладном аспектах; излагаются также прогнозы автора на будущее изучаемой научной области.

Таким образом, рецензируемая книга формирует и расширяет мировоззрение будущего выпускника вуза, повышая его рейтинг в жизни. В то же время книга является полноценным руководством для освоения современных технологий создания интеллектуальных систем и применения их для решения широкого круга практических про-

блем, встречающихся во многих областях деятельности современного человека

Книга рекомендуется в качестве учебника по изучению дисциплины «Интеллектуальные системы» для направления подготовки бакалавров по направлениям «Фундаментальная информатика и информационные технологии», «Информатика и вычислительная техника», «Информационные системы и технологии». Книга может быть полезна и для студентов других направлений, включающих изучение принципов создания интеллектуальных систем.

Рецензент:
Е. К. Хеннер,
*заведующий кафедрой информационных технологий
Пермского государственного национального
исследовательского университета,
доктор физ.-мат. наук, профессор.*

РЕЦЕНЗИЯ 2 НА КНИГУ Л. Н. ЯСНИЦКОГО «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ»

В России, как и во всем мире, не снижается спрос на IT-специалистов. Причем если раньше от них требовалось знание операционных систем, языков программирования и умение пользоваться пакетами прикладных программ, то теперь требуется знание методов интеллектуального анализа данных и умение их применять для решения практических задач.

Интеллектуальный анализ данных — это совокупность методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности. Наиболее эффективными из этих методов являются методы искусственного интеллекта, в частности — нейросетевые технологии.

Рецензируемая книга представляет собой учебник, построенный в классическом стиле. Он включает последовательное описание истории возникновения и развития научной области «Искусственный интеллект» как теоретической основы современных методов интеллектуального анализа данных. Раскрываются понятия данных и знаний, приводится описание наиболее распространенных способов представления знаний, а также основных технологий создания интеллектуальных систем. Сюда входят: технологии экспертных систем, эволюционного моделирования и технологии нейронных сетей, дается их сравнительный анализ. Основное внимание уделено нейросетевым технологиям, что обосновано современными успехами их приложений и все возрастающей популярностью этого математического аппарата среди IT-специалистов. Методически последовательно изложены теоретические основы, принципы проектирования и оптимизации нейронных сетей. Возможности нейронных сетей продемонстрированы на примерах их применения для интеллектуального анализа данных в промышленности, медицине, политологии, социологии, психологии. Очерчен круг решаемых задач, приведен рекомендуемый автором алгоритм применения метода нейросетевого моделирования.

В книге с единых позиций рассмотрены вопросы компьютерного творчества, причем изложение начинается с моделирования творческих процессов в гуманитарной сфере, после чего излагается авторский опыт создания интеллектуальных систем для решения краевых задач математической физики. Заканчивается книга сопоставлением различных технологий интеллектуального анализа данных и обсуждением проблем дальнейшего развития этой научной области.

Книга является исчерпывающим руководством по освоению современных технологий создания интеллектуальных систем и их применению для интеллектуального анализа данных широкого круга предметных областей. Ее можно рекомендовать в качестве учебника по изучению дисциплины «Интеллектуальные системы» для направления подготовки бакалавров 010300 «Фундаментальная информатика и информационные технологии»

Рецензент:
С. В. Русаков,
*заведующий кафедрой прикладной математики и информатики
Пермского государственного национального
исследовательского университета,
доктор физ.-мат. наук, профессор.*

Минимальные системные требования определяются соответствующими требованиями программы Adobe Reader версии не ниже 11-й для платформ Windows, Mac OS, Android, iOS, Windows Phone и BlackBerry; экран 10"

Учебное электронное издание

Серия: «Учебник для высшей школы»

Ясницкий Леонид Нахимович

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Учебник

Редактор *Т. Г. Хохлова*. Художник *В. Е. Шкерин*. Корректор *Д. И. Мурадян*
Оригинал-макет подготовлен *О. Г. Лапко* в пакете LaTeX 2ε

Подписано к использованию 15.06.16.
Формат 145×225 мм

Издательство «Лаборатория знаний»
125167, Москва, проезд Аэропорта, д. 3
Телефон: (499) 157-5272, e-mail: info@pilotLZ.ru, http://www.pilotLZ.ru



Автор учебника, **Леонид Нахимович Ясницкий**, — доктор технических наук, профессор Пермского государственного национального исследовательского университета, председатель Пермского отделения Научного совета Российской академии наук по методологии искусственного интеллекта, хорошо известен специалистам своими учебными пособиями, монографиями, научными работами, посвященными развитию и применению интеллектуальных информационных технологий.

Его новая книга написана с учетом быстро меняющихся реалий современной информатики. Возможности методов искусственного интеллекта продемонстрированы на примерах их применения в промышленности, экономике, медицине, политологии, социологии, психологии. Целый подраздел посвящен авторским know-how – технологическим секретам, которыми автор щедро делится с читателем. В заключение приводится сопоставление различных технологий интеллектуального анализа данных и обсуждение проблем дальнейшего развития этой научной области.

Книга позволяет получить необходимый минимум теоретических знаний, целостное представление о предмете и приобрести практические навыки разработки и применения интеллектуальных систем.